REPORT DOCUMENTATION PAGE	Form Approved OMB NO. 0704-0188

1. REPORT DATE (DD-MM-YYYY) 23-10-2017	2. REPORT TYPE Final Report	3. DATES COVERED (From - To) 18-Jul-2014 - 17-Jul-2017

4. TITLE AND SUBTITLE Final Report: Sociolinguistically Informed Natural Language Processing: Automating Irony Detection	5a. CONTRACT NUMBER W911NF-14-1-0442
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER 611102
6. AUTHORS	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES AND ADDRESSES University of Texas at Austin 101 East 27th Street Suite 5.300 Austin, TX 78712 -1532	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS (ES) U.S. Army Research Office P.O. Box 12211 Research Triangle Park, NC 27709-2211	10. SPONSOR/MONITOR'S ACRONYM(S) ARO 11. SPONSOR/MONITOR'S REPORT NUMBER(S) 66124-MA.5

12. DISTRIBUTION AVAILIBILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

The views, opinions and/or findings contained in this report are those of the author(s) and should not contrued as an official Department of the Army position, policy or decision, unless so designated by other documentation.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	15. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Byron Wallace
UU	UU	UU	UU		19b. TELEPHONE NUMBER 512-471-3821

RPPR Final Report
as of 26-Oct-2017

Agency Code:

Proposal Number: 66124MA | **Agreement Number: W911NF-14-1-0442**

INVESTIGATOR(S):

Name: Byron C. Wallace
Email: byron_wallace@brown.edu
Phone Number: 5124713821
Principal: Y

Name: David Beaver
Email: DIB@UTEXAS.EDU
Phone Number: 5124719028
Principal: N

Organization: **University of Texas at Austin**
Address: 101 East 27th Street, Austin, TX 787121532
Country: USA

DUNS Number: 170230239 | EIN: 746000203
Report Date: 17-Oct-2017 | Date Received: 23-Oct-2017
Final Report for Period Beginning 18-Jul-2014 and Ending 17-Jul-2017
Title: Sociolinguistically Informed Natural Language Processing: Automating Irony Detection
Begin Performance Period: 18-Jul-2014 | **End Performance Period:** 17-Jul-2017
Report Term: 0-Other
Submitted By: Byron Wallace | Email: byron_wallace@brown.edu
Phone: (512) 471-3821
Distribution Statement: 1-Approved for public release; distribution is unlimited.

STEM Degrees: **STEM Participants:**

Major Goals: As elaborated on at length in the attached (PDF) final report, the major goals of this project are summarized by the following aims:

Aim 1. To collect and annotate a high-quality corpus to facilitate research on irony detection. Prior to this project, no such high-quality dataset existed. This has been a major obstacle to progress on automated irony detection.

Aim 2. To analyze when existing ML and NLP technologies fail to detect ironic intent empirically. We specifically proposed to assess quantitatively (using the collected dataset) whether context is necessary to discern ironic intent (and how often this is the case).

Aim 3. Develop a new approach to irony detection that instantiates sociolinguistic conceptions of irony within a modern, probabilistic machine learning framework. This approach is to be informed by theoretical sociolinguistic perspectives on irony (and thus likely capable of discerning ironic utterances missed by existing computational models), while also being practical enough to be operational.

Accomplishments: Here we briefly summarize the major accomplishments with respect to each aim, but please see the attached final report for a detailed presentation.

With respect to Aim 1, we constructed the reddit irony corpus. This novel corpus has allowed to address unique questions and facilitated meaningful progress on this challenging task, both in our own work and by others (for example, a version of the corpus was added (by request) to the Kaggle platform (https://www.kaggle.com/rtatman/ironic-corpus)).

Concerning Aim 2, this work allowed us to ascertain that (1) humans require context to infer verbally ironic intent, and, (2) (standard) machine learning models tend fail more frequently on those cases for which humans require context. The latter observation subsequently motivated new work in this area on contextualized approaches to sarcasm detection, both by our group and by others, yielding meaningful progress.

Finally, toward realizing Aim 3, we have introduced novel strategies that exploit contextualizing information, that we have described in publications at top-tier venues. Some of this work was covered by New Scientist magazine (https://www.newscientist.com/article/2100007-ai-reads-your-tweets-and-spots-when-youre-being-sarc Finally, PI Beaver has recently made important progress on a conceptual framework that we believe will allow us to further advance approaches to verbal irony detection.

In sum, this project has resulted in: a novel annotated corpus for verbal irony detection; a new, quantified understanding of the importance of context in inferring verbal irony on behalf of speakers (afforded by the aforementioned corpus); and, finally, novel contextualized methods for automated sarcasm detection from online texts. These findings have been disseminated in four publications at top-tier NLP venues. According to Google Scholar all have been cited more than 10 times already (despite some being quite recent) and collectively they have been cited over 60 times (as of 10/1/2017). Furthermore, this work was featured in the ‘Tiny Transactions on Computer Science’ digest, which selectively highlights (in brief) cutting-edge work. Finally, PI Beaver has begun to develop an important theoretical work that will guide future work in the area. Thus, we strongly feel we have accomplished all project aims and that this work was a success. We provide greater detail of our findings and accomplishments in the remainder of this final report.

Training Opportunities: Nothing to Report

Results Dissemination: This paper directly supported the work described in the following publications:

Byron C. Wallace, Do Kook Choe, Laura Kertz, and Eugene Charniak. Humans require context to infer ironic intent (so computers probably do, too). In Proceedings of the Association for Computational Linguistics (ACL), pages 512–516. ACL, 2014.

Byron C Wallace, Do Kook, and Eugene Charniak. Sparse, Contextually Informed Models for Irony Detection: Exploiting User Communities, Entities and Sentiment. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL), pages 1035–1044, Beijing, China, 2015. ACL.

Ye Zhang, Stephen Roller, and Byron C. Wallace. MGNC-CNN: A Simple Approach to Exploiting Multiple Word Embeddings for Sentence Classification. In Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL), page 1522–1527. ACL, 2016.

Silvio Moreira, Byron C. Wallace, Hao Lyu, Paula Carvalho, and M ?ario J. Gaspar da Silva. Modelling Context with User Embeddings for Sarcasm Detection in Social Media. In Proceedings of the Conference on Computational Natural Language Learning (CoNLL), pages 167–177. SIGNLL, 2016.

Furthermore, we ran a workshop at CogSci, titled: Can cognitive scientists help computers recognize irony? in 2014. (see accompanying website: https://sites.google.com/a/brown.edu/irony/ and write-up: Byron C Wallace and Laura Kertz. Can cognitive scientists help computers recognize irony? In CogSci, 2014.)

The constructed corpus (dataset) has been shared publicly and one version of it is now featured on Kaggle, ensuring wide impact and dissemination: https://www.kaggle.com/rtatman/ironic-corpus.

Finally, the work was featured in the ‘Tiny Transactions on Computer Science’ digest, which selectively highlights (in brief) cutting-edge work.

Honors and Awards: Nothing to Report

Protocol Activity Status:

Technology Transfer: Nothing to Report

PARTICIPANTS:

Participant Type: PD/PI
Participant: Byron Casey Wallace

Person Months Worked: 9.00
Funding Support:
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Co PD/PI
Participant: David Beaver
Person Months Worked: 2.00
Funding Support:
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Co-Investigator
Participant: Laura Kertz
Person Months Worked: 2.00
Funding Support:
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Co-Investigator
Participant: Thomas Trikalinos
Person Months Worked: 2.00
Funding Support:
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

Participant Type: Co-Investigator
Participant: Eugene Charniak
Person Months Worked: 3.00
Funding Support:
Project Contribution:
International Collaboration:
International Travel:
National Academy Member: N
Other Collaborators:

CONFERENCE PAPERS:

RPPR Final Report
as of 26-Oct-2017

Publication Type: Conference Paper or Presentation **Publication Status:** 1-Published

Conference Name: CogSci

Date Received: Conference Date: Date Published:

Conference Location:

Paper Title: Can Cognitive Scientists Help Computers Recognize Irony?

Authors:

Acknowledged Federal Support:

Final Report: Sociolinguistically Informed Natural Language Processing: Automating Irony Detection (Proposal Number: 66124-MA, Agreement Number: W911NF1410442)

Scientific Progress and Accomplishments

David Beaver and Byron C. Wallace

2017-10

1 Background and Introduction

The research objective of this project was to develop novel computational methods to advance automated irony detection, i.e., identification of the ironic voice in online content (sarcasm detection constituting a key subset of verbal irony). This is a challenging task because the meaning of natural language is not captured by words and syntax alone. Rather, utterances (tweets,[1] sentences in forum posts, etc.) are embedded within a specific *context*. The ironic voice is an important example of this phenomenon: to appreciate a speaker's intended meaning, it is crucial to first infer if he or she is being ironic or sincere.

Earlier computational approaches to irony detection have leveraged essentially standard statistical natural language processing (NLP) and machine learning (ML) methods. These models tend to be relatively 'shallow' in that they operate only over simple, unstructured representations of data. For example, in the case of natural language (text), one might encode documents with word counts or functions thereof, and in the case of network-based data (e.g. social networks) one might rely on analogously simple functions of link counts. Classification may then be performed by algorithms operating over these encodings. However, these simple representations will often be insufficient to infer ironic intent [33].

This project has allowed us to demonstrate empirically that context is necessary to infer ironic intent [36]. And more recently this support has enabled us to develop novel models that exploit contextual cues to better inform predictions [35, 20]. We have also recently been exploring innovative modern neural network variants that capitalize on richer representations of text [38]. We elaborate on this progress below. Together, these contributions (described in papers published at top natural language processing conferences) constitute substantial progress, as we elaborate on below.

For this project, we brought together a diverse team comprising members with unique expertise. Senior personnel on this project includes PI Wallace (formerly at UT Austin and now transitioning to Northeastern University) and Eugene Charniak (Brown University), both of whom have substantive expertise in statistical natural language processing. Also involved is David Beaver (UT Austin), a Professor of Linguistics and the head of the Cognitive Science Program (`http://www.utexas.edu/cola/linguistics/faculty/profile.php?id=dib97`). Professor Beaver brought a unique philosphical perspective to bear on the project and took over the project from fall 2016 onward, as Wallace transitioned to Northeastern University. Although no longer formally involved in the project, Wallace maintained contact with Beaver throughout this final year.

The interdisciplinary team just described has been an important aspect of our approach. In our view previous efforts to identify irony relied too heavily on standard computer science methods, largely ignoring

[1] 'tweets' are short messages posted to the internet for the consumption of 'followers' via the web service Twitter.

the perspectives of, e.g., linguistics and cognitive scientists. Indeed, as part of this broad effort of facilitating interdisciplinary communication around this important problem, we organized and ran a workshop at CogSci 2014, which included speakers and attendees from both computer and cogntive science (`https://sites.google.com/a/brown.edu/irony/`; [34]).

In the next section we review the specific objectives of this project and then briefly the ways in which we have achieved them. The remainder of the document provides additional details.

2 Project Accomplishments Summary and Highlights

2.1 Specific Objectives

Below we review what the general objectives of this project were and then specific accomplishments we have accomplished with respect to these (which we later elaborate on in detail).

- **Aim 1**. *To collect and annotate a high-quality corpus to facilitate research on irony detection.* Prior to this project, no such high-quality dataset existed. This has been a major obstacle to progress on automated irony detection.

  Accomplishments. We constructed the reddit irony corpus [36]. This novel corpus has allowed to address unique questions and facilitated meaningful progress on this challenging task, both in our own work and by others (for example, a version of the corpus was added (by request) to the Kaggle platform (`https://www.kaggle.com/rtatman/ironic-corpus`)).

- **Aim 2**. *To analyze when existing ML and NLP technologies fail to detect ironic intent* empirically. We specifically proposed to assess quantitatively (using the collected dataset) whether *context* is necessary to discern ironic intent (and how often this is the case).

  Accomplishments. As we summarize in the results below, this work allowed us to ascertain that (1) humans require context to infer verbally ironic intent, and, (2) (standard) machine learning models tend fail more frequently on those cases for which humans require context. The latter observation subsequently motivated new work in this area on *contextualized* approaches to sarcasm detection, both by our group [35, 20] and by others [2], yielding meaningful progress.

- **Aim 3**. *Develop a new approach to irony detection that instantiates sociolinguistic conceptions of irony within a modern, probabilistic machine learning framework.* This approach is to be informed by theoretical sociolinguistic perspectives on irony (and thus likely capable of discerning ironic utterances missed by existing computational models), while also being practical enough to be operational.

  Accomplishments. As we describe at length below, we have introduced novel strategies that exploit contextualizing information [35, 20, 38]. Some of this work was covered by New Scientist magazine (`https://www.newscientist.com/article/2100007-ai-reads-your-tweets-and-spots-when-youre-being-sarc` Finally, PI Beaver has recently made important progress on a conceptual framework that we believe will allow us to further advance approaches to verbal irony detection.

In sum, this project has resulted in: a novel annotated corpus for verbal irony detection; a new, quantified understanding of the importance of *context* in inferring verbal irony on behalf of speakers (afforded by the aforementioned corpus); and, finally, novel contextualized methods for automated sarcasm detection from online texts. These findings have been disseminated in four publications at top-tier NLP venues [36, 35, 20, 38]. According to Google Scholar all have been cited more than 10 times already (despite some being quite recent) and collectively they have been cited over 60 times (as of 10/1/2017). Furthermore, this work was featured in the 'Tiny Transactions on Computer Science' digest, which selectively highlights (in brief) cutting-edge work [15]. Finally, PI Beaver has begun to develop an important theoretical work that will guide future work in the area. Thus, we strongly feel we have accomplished all project aims and that this work was a success. We provide greater detail of our findings and accomplishments in the remainder of this final report.

3 Scientific Findings and Accomplishments

As just reviewed, we have achieved all project aims. This has culminated in four publications in total over the project period [36, 35, 20, 38]. Specifically, as described in detail in the following subsections, we have: written code to scrape comments from *reddit*, a social-news website that we use as our corpus; built a web-based tool to facilitate annotation of these comments; assembled and trained a team of undergraduates to perform this annotation; analyzed the resultant dataset. This analysis was summarized in our publication in the proceedings of the 2014 Association for Computational Linguistics (ACL) [36], one of the highest quality venues in natural language processing. This analysis has since at least partially motivated new, context-aware approaches to irony detection by other researches [2], which was one of our aims.

Using this corpus, we were then able to make significant progress on building new machine learning models that exploit context and richer representations to improve identification of verbal irony. In particular, we developed a new model for irony detection that exploits contextual cues to improve classification performance [35], also published in ACL (2015). Furthermore, we have developed novel neural models (and in particular, Convolutional Neural Networks, or CNNs) to better capitalize on distributed representations of words in content, thus providing additional linguistic context [38]. Most recently, we published a method that further extends the neural approach to *learn a distributed representation of individuals* that then informs the prediction as to whether or not they are being ironic [20]. We note that the latter publication has received attention from major technology media outlets, including (as noted above) New Scientist and TechCrunch.[2]

In the final year of this project, Prof. Beaver took over as PI. In the past year, Beaver has been developing a general framework for studying these issues that we can use to contextualize and inform models of irony moving forwards, has presented preliminary results in multiple recent colloquia (Yale University, the University of Chicago, and the Leibniz-Zentrum Allgemeine Sprachwissenschaft in Berlin; and he will present the full model in a jointly authored book "Politics of Language" under contract with Princeton University Press, ms. to be delivered to publisher 1/1/2018). Below we summarize in detail how the conceptual model PI Beaver has developed applies within the project. Additionally, we note that Beaver worked with a graduate student, Luke Pinnette, performing a semi-independent replication analysis of the method and results that the team reported in Wallace et al (2014) and Wallace et al (2015). Pinnette produced a technical report *Speaker History and Other Cues to Irony in a Computational Model*, which explores features and approaches that might be used in future irony detection work. Pinnette was due to develop this project further in Spring/Summer 2017, but was unfortunately forced to leave the graduate program and project due to circumstances beyond our control.

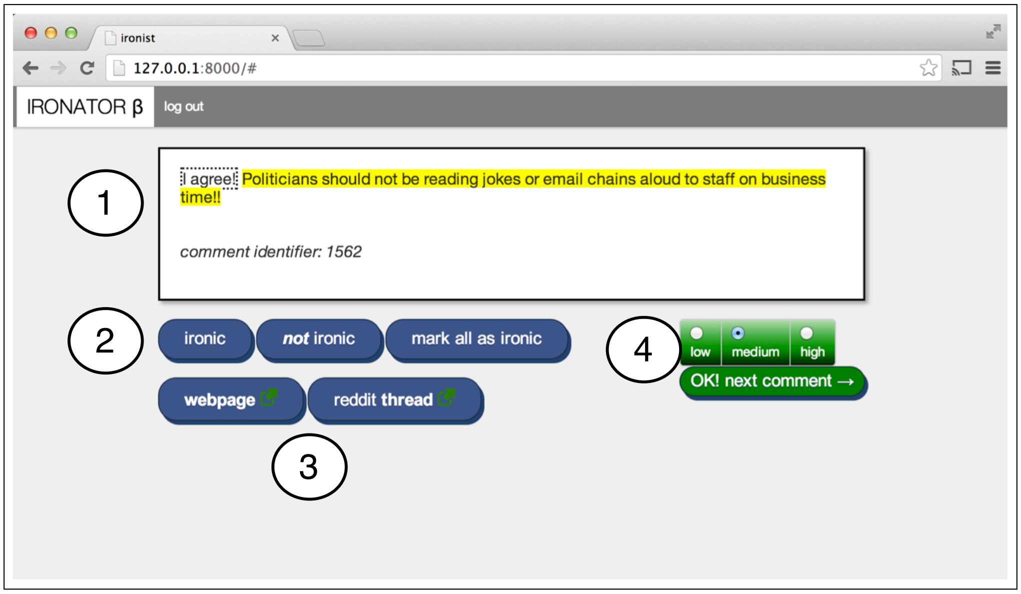
In the remainder of this report, we elaborate on the findings and outcomes summarized above.

3.1 The reddit Irony Dataset

One important outcome of this project was the new Reddit corpus. Reddit (`http://reddit.com`) is a social-news website to which news stories (and other links) are posted, voted on and commented upon. The forum component of reddit is extremely active: popular posts often have well into 1000's of user comments. Reddit comprises 'sub-reddits', which focus on specific topics. For example, `http://reddit.com/r/politics` features articles (and hence comments) centered around political news. The current version of the corpus is available at: `https://github.com/bwallace/ACL-2014-irony`. The present version comprises 3,020 annotated comments scraped from the six subreddits enumerated in Table 1. These comments in turn comprise a total of 10,401 labeled sentences.[3] A version of this dataset has now been added (by request) to the Kaggle repository, ensuring maximal impact.

[2] `https://techcrunch.com/2016/08/04/this-neural-network-tries-to-tell-if-youre-being-sarcastic-online/`

[3] We performed naïve 'segmentation' of comments based on punctuation.


I agree!! Politicians should not be reading jokes or email chains aloud to staff on business time!!

comment identifier: 1562

Figure 1: The web-based tool used by our annotators to label reddit comments. Enumerated interface elements are described as follows: **1** the text of the comment to be annotated – sentences marked as *ironic* are highlighted; **2** buttons to label sentences as *ironic* or *unironic*; **3** buttons to request additional *context* (the embedding discussion thread or associated webpage – see Section 3.1.2); **4** radio button to provide *confidence* in comment labels (Likert scale of *low*, *medium* and *high*).

3.1.1 Annotation Process

Three Brown university undergraduates independently annotated each sentence in the corpus. More specifically, annotators have provided binary 'labels' for each sentence indicating whether or not they (the annotator) believe it was intended by the author ironically (or not). This annotation was facilitated via a custom-built browser-based annotation tool developed as part of this project, shown in Figure 1.

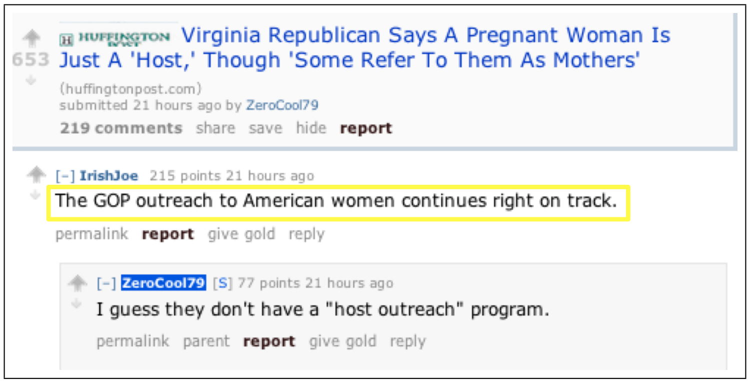
We intentionally did not provide much guidance to annotators regarding the criteria for what constitutes an 'ironic' statement, for two reasons. First, verbal irony is a notoriously slippery concept [12] and coming up with an operational definition to be consistently applied is non-trivial. Second, we were interested in assessing the extent of natural agreement between annotators for this task. The raw average agreement between all annotators on all sentences is 0.844. Average pairwise Cohen's Kappa [8] is 0.341, suggesting fair to moderate agreement [32], as we might expect for a subjective task like this one. Nonetheless, ideally we would perhaps achieve better agreement; large disagreements like this are suboptimal for NLP models. Future work might thus re-visit issues of annotator agreement. In general, we feel better understanding annotator (dis-)agreement in subjective natural language processing tasks is an important and under-studied aim. At the very least, our present results indicate an upper-bound for what we can possibly expect from an automated approach).

3.1.2 Context

Reddit is an ideal corpus for the irony detection task in part because it provides a natural practical realization of the otherwise ill-defined *context* for comments (and the sentences they comprise). In particular, each comment is associated with a specific user (the author), and we can view their previous comments. Moreover, comments are embedded within discussion *threads* that pertain to the (usually external) content linked to in the corresponding submission (see Figure 2). These pieces of information (previous comments by the same user, the external link of the embedding reddit thread, and the other comments in this thread) constitute

sub-reddit (URL)	description	number of labeled comments
politics (r/politics)	Political news and editorials; focus on the US.	873
conservative (r/conservative)	A community for political conservatives.	573
progressive (r/progressive)	A community for political progressives (liberals).	543
atheism (r/atheism)	A community for non-believers.	442
Christianity (r/Christianity)	News and viewpoints on the Christian faith.	312
technology (r/technology)	Technology news and commentary.	277

Table 1: The six sub-reddits that we have downloaded comments from and the respective numbers of which we have acquired annotations. Note that we acquired labels at the *sentence* level, whereas the counts above reflect *comments*, all of which contain at least one sentence.



653 HUFFINGTON POST Virginia Republican Says A Pregnant Woman Is Just A 'Host,' Though 'Some Refer To Them As Mothers'

(huffingtonpost.com)
submitted 21 hours ago by ZeroCool79

[–] IrishJoe 215 points 21 hours ago

The GOP outreach to American women continues right on track.

[–] ZeroCool79 [S] 77 points 21 hours ago

I guess they don't have a "host outreach" program.

Figure 2: An illustrative reddit comment (highlighted). The title ("Virginia Republican ...") links to an article, providing one example of contextualizing content. The conversational thread in which this comment is embedded provides additional context. The comment in question was presumably intended ironically, but without the aforementioned context this would be difficult to conclude with any certainty. Because all of this information is readily available online, we think automated approaches ought to try and exploit it.

our context. All of this is readily accessible. Labelers can opt to request these pieces of context via the annotation tool, and we record when they do so.

Consider the following example comment taken from our dataset: "Great idea on the talkathon Cruz. Really made the republicans look like the sane ones." Did the author intend this statement ironically, or was this a subtle dig on Senator Ted Cruz? Without additional context it is difficult to know. And indeed, all three annotators requested additional context for this comment. This context at first suggests that the comment may have been intended literally: it was posted in the r/conservative subreddit (Ted Cruz is a conservative senator). But if we peruse the author's comment history, we see that he or she repeatedly derides Senator Cruz (e.g., writing "Ted Cruz is no Ronald Reagan. They aren't even close."). From this contextual information, then, we can reasonably assume that the comment was intended ironically (and all three annotators did so after assessing the available contextual information).

3.2 Humans Need Context to Infer Irony

We explore the extent to which human annotators rely on contextual information to decide whether or not sentences were intended ironically. Recall that our annotation tool allows labelers to request additional context if they cannot make a decision based on the comment text alone (Figure 1). On average, annotators requested additional context for 30% of comments (range across annotators of 12% to 56%). As shown in Figure 3, annotators are consistently more confident once they have consulted this information.

We tested for a correlation between these requests for context and the final decisions regarding whether comments contain at least one ironic sentence. We denote the probability of at least one annotator requesting

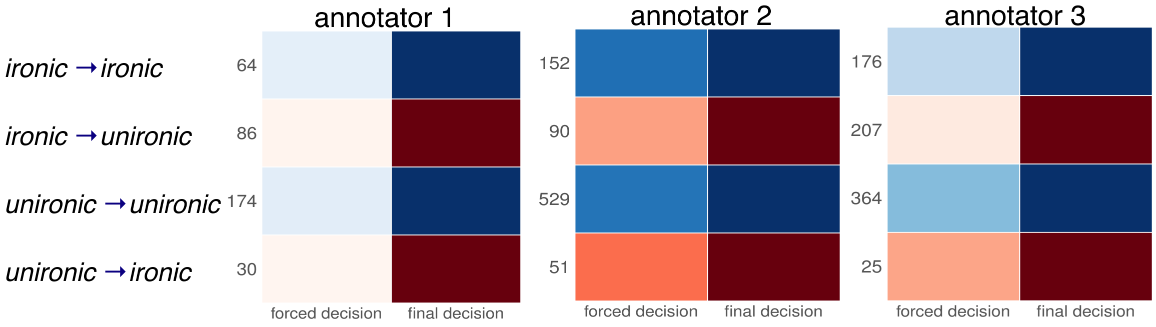


Figure 3: This plot illustrates the effect of viewing contextual information for three annotators (one table for each annotator). For all comments for which these annotators requested context, we show *forced* (before viewing the requested contextual content) and *final* (after) decisions regarding perceived ironic intent on behalf of the author. Each row shows one of four possible decision sequences (e.g., a judgement of *ironic* prior to seeing context and *unironic* after). Numbers correspond to counts of these sequences for each annotator (e.g., the first annotator changed their mind from *ironic* to *unironic* 86 times). Cases that involve the annotator changing his or her mind are shown in red; those in which the annotator stuck with their initial judgement are shown in blue. Color intensity is proportional to the average confidence judgements the annotator provided: these are uniformly stronger after they have consulted contextualizing information. Note also that the context frequently results in annotators changing their judgement.

additional context for comment i by $P(\mathcal{C}_i)$. We then model the probability of this event as a linear function of whether or not any annotator labeled any sentence in comment i as ironic. We code this via the indicator variable $\mathcal{I}_i$ which is 1 when comment i has been deemed to contain an ironic sentence (by any of the three annotators) and 0 otherwise.

$$logit\{P(\mathcal{C}_i)\} = \beta_0 + \beta_1 \mathcal{I}_i \tag{1}$$

We used the regression model shown in Equation 1, where β_0 is an intercept and β_1 captures the correlation between requests for context for a given comment and its ultimately being deemed to contain at least one ironic sentence. We fit this model to the annotated corpus, and found a significant correlation: $\hat{\beta}_1 = 1.508$ with a 95% confidence interval of (1.326, 1.690); $p < 0.001$.

In other words, annotators request context significantly more frequently for those comments that (are ultimately deemed to) contain an ironic sentence. This would suggest that the words and punctuation comprising online comments alone are not sufficient to distinguish ironic from unironic comments. Despite this, most machine learning based approaches to irony detection have relied nearly exclusively on such intrinsic features.

3.3 Machines Probably do, too

To address research objective 2 above, we explored whether the misclassifications (with respect to whether comments contain irony or not) made by a standard text classification model significantly correlate with those comments for which human annotators requested additional context. It turns out that it does. This provides evidence that bag-of-words approaches are insufficient for the general task of irony detection: more context is necessary.

Specifically, we implemented a baseline classification approach using vanilla token count features (binary bag-of-words). We removed stop-words and limited the vocabulary to the 50,000 most frequently occurring unigrams and bigrams. We added additional binary features coding for the presence of punctuational features, such as exclamation points, emoticons (for example, ';)') and question marks: previous work [10, 6] has found that these are good indicators of ironic intent.

For our predictive model, we used a linear-kernel SVM (tuning the C parameter via grid-search over the training dataset to maximize F1 score). We performed five-fold cross-validation, recording the predictions $\hat{y}_i$ for each (held-out) comment i. Average F1 score over the five-folds was 0.383 with range (0.330, 0.412); mean recall was 0.496 (0.446, 0.548) and average precision was 0.315 (0.261, 0.380). The five most predictive tokens were: *!*, *yeah*, *guys*, *oh* and *shocked*. This represents reasonable performance (and the high ranking tokens are as expected); but obviously there is quite a bit of room for improvement.

We now explore empirically whether the these misclassifications are made on the same comments for which annotators requested context. To this end, we introduce a variable $\mathcal{M}_i$ for each comment i such that $\mathcal{M}_i = 1$ if $\hat{y}_i \neq y_i$, i.e., $\mathcal{M}_i$ is an indicator variable that encodes whether or not the classifier misclassified comment i. We then ran a second regression in which the output variable was the logit-transformed probability of the model misclassifying comment i, i.e., $P(\mathcal{M}_i)$. Here we are interested in the correlation of the event that one or more annotators requested additional context for comment i (denoted by $\mathcal{C}_i$) and model misclassifications (adjusting for the comment's true label). Formally:

$$logit\{P(\mathcal{M}_i)\} = \theta_0 + \theta_1 \mathcal{I}_i + \theta_2 \mathcal{C}_i \tag{2}$$

Fitting this to the data, we estimated $\hat{\theta}_2 = 0.930$ with a 95% CI of (0.769, 1.093); $p < 0.001$. Put another way, the model makes mistakes on those comments for which annotators requested additional context (even after accounting for the annotator designation of comments). This motivates our subsequent work (described below) on operationalizing context to reduce mistakes.

3.4 Sparse, Contextually Informed Models for Irony Detection: Exploiting User Communities, Entities and Sentiment

Motivated by the findings just reported, we developed a new model that capitalizes on contextual information to improve irony detection [35]. Specifically, this model combinines noun phrases and sentiment extracted from comments with the sub-reddit type (e.g., conservative or liberal) to which they were posted. Because this method generates a very large feature space (and we expect predictive contextual features to be strong but few), we have proposed a mixed regularization strategy that places a sparsity-inducing ℓ_1 penalty on the contextual feature weights on top of the ℓ_2 penalty applied to all model coefficients. This increases model sparsity and reduces the variance of model performance.

As discussed above, previous models for irony detection [30, 18, 25] have relied predominantly on features *intrinsic* to the texts to be classified. By contrast, we propose exploiting *contextualizing* information, which is often available for web-based classification tasks. More specifically, we exploit signal gleaned from the conversational *threads* to which comments belong. Our approach capitalizes on the intuition that members of different user communities are likely to be sarcastic about different things. As a proxy for user community, we leverage knowledge of the specific forums to which comments were posted. For example, one may surmise that the statement 'I really am proud of Obama' is likely to have been intended ironically if it was posted to a forum frequented by political conservatives. But if this same utterance were posted to a liberal-leaning forum, it is more likely to have been intended in earnest. This sort of information is often directly or indirectly available on social media, but previous models have not capitalized on it. This is problematic, as discussed above (and as we report in [36]).

To evaluate this contextually aware approach, we consider comments posted to two pairs of polarized user communities, or subreddits: (1) *progressive* and *conservative* subreddits (comprising individuals on the left and right of the US political spectrum, respectively), and (2) *atheism* and *Christianity* subreddits. Using these datasets, we have made the following research contributions:

- We have shown that contextual information, such as inferred user-community (in this case, the sub-reddit) can be crossed with extracted entities and sentiment to improve detection of verbal irony to improve performance over baseline models (including those that exploit inferred sentiment, but not context).

Feature	Description
Sentiment	The inferred sentiment (*negative/neutral* or *positive*) for a given comment.
Subreddit	the subreddit (e.g., *progressive* or *conservative*; *atheism* or *Christianity*) to which a comment was posted.
NNP	Noun phrases (e.g., proper nouns) extracted from comment texts.
NNP+	Noun phrases extracted from comment texts *and* the thread to which they belong.

Table 2: Feature types that we exploit. We view the (observed) subreddit as a proxy for *user type*. We combine this with sentiment and extracted noun phrases (NNPs) to improve classifier performance.

- We propose a novel composite regularization strategy that applies a sparsifying ℓ_1 penalty to the contextual/sentiment/entity feature weights in addition to the standard squared ℓ_2 penalty to all feature weights. This induces more compact, interpretable models that exhibit lower variance.

The motivation for our model derives from the large body of work on the use and interpretation of verbal irony supports the supposition that context plays a critical role in discerning verbal irony [13, 7, 33, 36]. Individuals will be more likely, in general, to use sarcasm when discussing specific entities. Which entities will depend in part on the community to which the individual belongs. As a proxy for user community, here we leverage the subreddits to which comments were posted. Sentiment may also play an important role. In general, verbal irony is almost always used to convey negative views via ostensibly positive utterances [28]. And recent work [25] has exploited features based on sentiment to improve irony detection.

To summarize: when assuming an ironic voice we expect that individuals will convey ostensibly positive sentiment about entities, and that these entities will depend on the type of individual in question. We propose capitalizing on such information by introducing features that encode subreddits, sentiment and noun phrases (NNPs), as we describe next.

3.4.1 Features

We leverage the feature sets enumerated in Table 2. Subreddits are observed variables. Noun phrase (NNP) extraction and sentiment inference are performed automatically via state of the art NLP tools. In particular, we use the Stanford Sentiment Analysis tool [27] to infer sentiment. To extract NNPs we use the Stanford Part of Speech tagger [29]. We then introduce 'bag-of-NNP' features and features that indicate whether the sentiment inferred for a given sentence was positive or not.

Additionally, we introduce 'interaction' features that capture combinations of these. For example, a feature that indicates whether a given sentence mentions Obamacare (which will be one of many NNPs automatically extracted) *and* was posted in the *conservative* subreddit. This is an example of a two-way interaction. We also experiment with three-way interactions, crossing sentiment with NNPs and subreddits. An example is a feature that indicates if a sentence was: inferred to be positive *and* mentions Obamacare (NNP) *and* was part of a comment made in the conservative subreddit. Finally, we experiment with adding NNPs extracted from the comment thread in addition to the comment text.

These are rich features that capture signal not directly available from the sentences themselves. Features that encode subreddits crossed with extracted NNP's, in particular, offer a chance to explicitly account for differences in how the ironic device is used by individuals in different communities. However, this has the downside of introducing a large number of irrelevant terms into the model: we expect, *a priori*, that many entities will not correlate with the use of verbal irony. We would therefore expect this strategy to exhibit high variance in terms of predictive performance, and we later confirm this empirically. Ideally, a model would perform feature selection during parameter estimation, thus dropping irrelevant interaction terms. We next introduce a composite ℓ_1/ℓ_2 regularization strategy toward this end.

3.4.2 Enforcing sparsity

Here we consider linear models with binary outputs ($y \in \{-1, +1\}$). We will assume we have access to a training dataset comprising n instances, $\mathbf{x} = \{\mathbf{x}_1, ..., \mathbf{x}_n\}$ and associated labels $\mathbf{y} = \{y_1, ..., y_n\}$. We then aim to find a weight-vector $\mathbf{w}$ that optimizes the following objective.

$$\mathbf{w} \sum_{i=1}^{n} \mathcal{L}(\text{sign}\{\mathbf{w} \cdot \mathbf{x}_i\}, y_i) + \alpha \mathcal{R}(\mathbf{w}) \tag{3}$$

Where $\mathcal{L}$ is a loss function, $\mathcal{R}(\mathbf{w})$ is a regularization term and α is a parameter expressing the relative emphasis placed on achieving minimum empirical loss versus producing a simple model (i.e., a weight vector with small weights). Typically one searches for a good α using the available training data. For $\mathcal{L}$, we will use the log-loss in this work, though other loss functions may be used in its place.

Concerning $\mathcal{R}$, one popular regularization function is the squared ℓ_2 norm:

$$\sum_j \mathbf{w}_j^2 \tag{4}$$

This is the norm used in the standard Support Vector Machine (SVM) formulation, for example, and has been shown empirically to work well for text classification. An alternative is to use the ℓ_1 norm:

$$\sum_j |\mathbf{w}_j| \tag{5}$$

Which has the advantage of inducing sparse models: i.e., using the ℓ_1 norm as a penalty tends to drive feature weights to 0.

Returning to the present task of detecting verbal irony in comments, it seems reasonable to assume that there will be a relatively small set of entities that correlate with sarcasm. But because we are introducing 'interaction' features that enumerate the cross-product of subreddits and entities (and, in some cases, sentiment), we have a large feature-space. This space includes features that correspond to NNPs extracted from, and sentiment inferred for, the sentence itself: we will denote the indices for these by $\mathcal{I}$. Other interaction features correspond to entities extracted from the *threads* associated with comments: we denote the corresponding set of indices by $\mathcal{T}$. We expect only a fraction of the features comprising both $\mathcal{I}$ and $\mathcal{T}$ to have non-zero weights (i.e., to signal ironic intent).

This scenario is prone to the undesirable property of high-variance, and hence calls for stronger regularization. But in general replacing the squared ℓ_2 norm with an ℓ_1 penalty (over all weights) hampers classification performance (indeed, as we later report, this strategy performs very poorly here). Therefore, in our scenario we would like to place a sparsifying ℓ_1 regularizer over the contextual (interaction) features while still leveraging the squared ℓ_2-norm penalty for the standard bag-of-words (BoW) features.[4] We thus propose the following composite penalty:

$$\sum_j \mathbf{w}_j^2 + \sum_{k \in \mathcal{I}} |\mathbf{w}_k| + \sum_{l \in \mathcal{T}} |\mathbf{w}_l| \tag{6}$$

The idea is that this will drive many of the weights associated with the contextual features to zero, which is desirable in light of the intuition that a relatively small number of entities will likely indicate sarcasm. At the same time, this composite penalty applies only the squared ℓ_2 norm to the standard BoW features, given the comparatively strong predictive performance realized with this strategy.

Putting this together, we modify the original objective (Equation 3) as follows:

$$\mathbf{w} \sum_{i=1}^{n} \mathcal{L}(\text{sign}\{\mathbf{w} \cdot \mathbf{x}_i\}, y_i) + \alpha_0 \sum_j \mathbf{w}_j^2 + \alpha_1 \sum_{k \in \mathcal{I}} |\mathbf{w}_k| + \alpha_2 \sum_{l \in \mathcal{T}} |\mathbf{w}_l| \tag{7}$$

Where we have placed separate α scalars on the respective penalty terms. Note that this is similar to the *elastic net* [40] joint regularization and variable selection strategy. The distinction here is that we only apply the ℓ_1 penalty to (i.e., perform feature selection for) the subset of 'interaction' feature weights, which is in

[4] Note that we apply both ℓ_1 and ℓ_2 penalties to the features in $\mathcal{I}$ and $\mathcal{T}$.

contrast to the elastic net, which imposes the composite penalty to *all* feature weights. One can view this as using the regularizer to encourage a sparsity pattern specific to the task at hand.

We fit this model via Stochastic Gradient Descent (SGD). During each update, we impose both the squared ℓ_2 and ℓ_1 penalties; the latter is applied only to the contextual/interaction features in $\mathcal{I}$ and $\mathcal{T}$. For the ℓ_1 penalty, we adopt the cumulative truncated gradient method proposed by Tsuruoka et al. [31].

3.5 Experimental Setup and Results

3.5.1 Datasets

We now report empirical results concerning the performance of the model just sketched. We use datasets derived from our reddit corpus, described in detail above (Section 3.1). More specifically, for our development dataset, we used a subset of this corpus comprising annotated comments from the *progressive* and *conservative* subreddits. We also report results from experiments performed using a separate, held-out portion of this data, which we did not use during model refinement. Furthermore, we later present results on comments from the *atheism* and *Christianity* subreddits (we did not use this data during model development, either).

The development dataset includes 1,825 annotated comments (876 and 949 from the *progressive* and *conservative* subreddits, respectively). These comprise 5,625 sentences in total, each of which was independently labeled by three annotators as having been intended *ironically* or not, as described in Section 3.1.1. For simplicity, we consider a sentence to be 'ironic' ($y = 1$) when at least two of the three annotators designated it as such, and 'unironic' ($y = -1$) otherwise. Using this criteria, 286 (5%) of the labeled sentences are labeled 'ironic'.

The test portion of the political dataset comprises 996 annotated comments (409 *progressive* and 587 *conservative* comments), totalling 2,884 sentences. Using the same criteria as above – at least 2/3 annotators labeling a given sentence as 'ironic' – we have 154 'ironic' sentences (again about 5%).

The 'religion' dataset (comments from *atheism* and *Christianity*) contains 1,682 labeled comments comprising 5615 sentences (2,966 and 2,649 from the atheism and Christian subreddits, respectively); 313 (∼6%) were deemed 'ironic'.

We recorded results from 500 independently performed experiments on random train (80%)/test (20%) splits of the data. These splits were performed at the *comment* (rather than sentence) level, so as not to test on sentences belonging to comments encountered in the training set. We measured performance, however, at the sentence level (often only a single sentence in a given comment will have been labeled as 'ironic').

Our baseline approach is a standard squared-ℓ_2 regularized log-loss linear model (fit via SGD) that leverages uni- and bi-grams and features indicating grammatical cues, such as exclamation points and emoticons. We also experiment with a model that includes inferred sentiment indicators, but not context. We performed standard English stopwording, and we used Term Frequency Inverse-Document Frequency (TF-IDF) feature weighting. For the gradient descent procedure, we used a decaying learning rate (specifically, $\frac{1}{t}$, where t is the update count). We performed a coarse grid search to find values for α that maximize $F1$ on the training datasets. We took five full passes over the training data before terminating descent.

We report paired *recalls* and *precisions*, as observed on each random train/test split of the data. The former is defined as $\frac{TP}{TP+FN}$ and the latter as $\frac{TP}{TP+FP}$, where TP denotes the true positive count, FN the number of false negatives and FP the false positive count. We report these separately - rather than collapsing into $F1$ - because it is not clear that one would value recall and precision equally for irony detection, and because this allows us to tease out *how* the models differ in performance. Notably, for example, sentiment and context features both improve recall, but the latter does so without harming precision.

3.5.2 Results

Figure 4 and Table 3 summarize the performance of the different approaches over 500 independently performed train/test splits of the political development corpus. For reference, a random chance strategy (which predicts 'ironic' with probability equal to the observed prevalence) achieves a median recall of 0.048 and a median precision of 0.047.

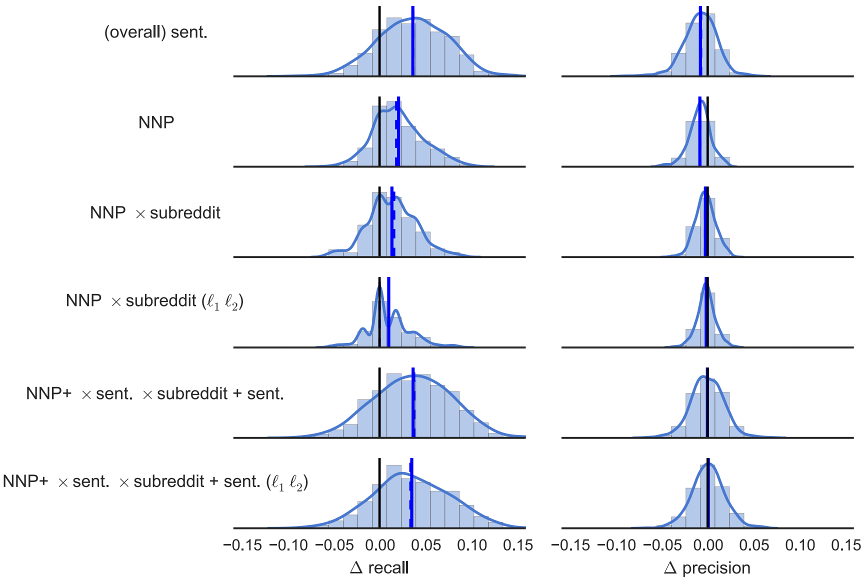


Figure 4: Results from 500 independent train/test splits of the development subset of our political data. Shown are histograms with smoothed kernel density estimates of differences in recall and precision between the baseline bag-of-words based approach and each feature space/method (one per row). The solid black line at 0 indicates no difference; solid and dotted blue lines demarcate means and medians, respectively. Features are as in Table 2. The × symbol denotes interactions; + indicates addition. The proposed contextual features substantially improve recall, with little to no loss in precision. Moreover, in general, the $\ell_1\ell_2$ regularization approach reduces variance. (We note that in constructing histograms we have excluded a handful of points – never more than 1% – where the difference exceeded 0.15).

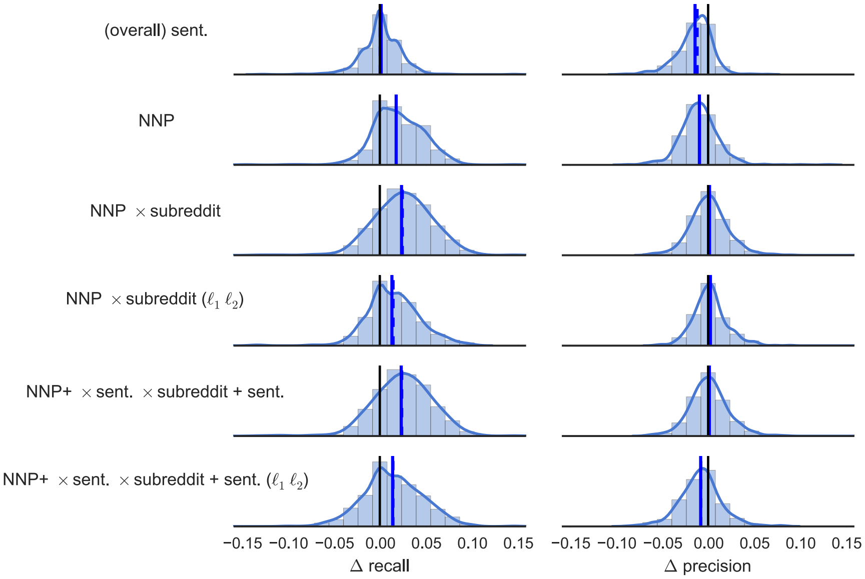


Figure 5: Results from 500 independent train/test splits of the development subset of the religion corpus). The description is the same as for Figure 4.

	mean; median (25th, 75th)	mean; median (25th, 75th)
baseline (BoW)	0.288; 0.283 (0.231, 0.333)	0.129; 0.124 (0.103, 0.149)
	Δ recall	Δ precision
(overall) sent.	+0.036; +0.037 (+0.015, +0.063)	-0.008; -0.007 (-0.018, +0.003)
NNP	+0.021; +0.018 (+0.000, +0.036)	-0.008; -0.008 (-0.016, -0.001)
NNP $\times$ subreddit	+0.013; +0.016 (+0.000, +0.031)	-0.002; -0.003 (-0.009, +0.004)
NNP $\times$ subreddit (ℓ_1 ℓ_2)	+0.010; +0.000 (+0.000, +0.021)	-0.002; -0.002 (-0.007, +0.004)
NNP+ $\times$ sent. $\times$ subreddit + sent.	+0.036; +0.038 (+0.000, +0.065)	-0.000; -0.001 (-0.012, +0.011)
NNP+ $\times$ sent. $\times$ subreddit + sent. (ℓ_1 ℓ_2)	+0.035; +0.034 (+0.000, +0.062)	+0.001; +0.000 (-0.011, +0.011)

Table 3: Summary results over 500 random train/test splits of the development dataset. The top row reports mean and median baseline (BoW) recall and precision and lower and upper (25th and 75th) percentiles. We report pairwise differences w.r.t. this baseline in terms of recall and precision for each strategy. Exploiting NNP features and subreddits improves recall with little to not cost in precision. Capitalizing on sentiment alone improves recall but at a greater cost in precision. The proposed $\ell_1\ell_2$ regularization strategy achieves comparable performance with fewer features, and shrinks the variance over different train/test splits (as can bee seen in Figure 4).

	mean; median (25th, 75th)	mean; median (25th, 75th)
baseline (BoW)	0.281; 0.268 (0.222, 0.327)	0.189; 0.187 (0.144, 0.230)
	Δ recall	Δ precision
(overall) sent.	+0.001; +0.000 (-0.011, +0.015)	-0.014; -0.012 (-0.023, -0.002)
NNP	+0.018; +0.018 (+0.000, +0.039)	-0.009; -0.010 (-0.021, +0.001)
NNP $\times$ subreddit	+0.024; +0.025 (+0.000, +0.046)	+0.002; +0.001 (-0.011, +0.013)
NNP $\times$ subreddit (ℓ_1 ℓ_2)	+0.013; +0.015 (+0.000, +0.033)	+0.002; +0.002 (-0.009, +0.011)
NNP+ $\times$ sent. $\times$ subreddit + sent.	+0.023; +0.024 (+0.000, +0.046)	+0.001; +0.001 (-0.012, +0.013)
NNP+ $\times$ sent. $\times$ subreddit + sent. (ℓ_1 ℓ_2)	+0.014; +0.015 (+0.000, +0.036)	-0.008; -0.008 (-0.021, +0.004)

Table 4: Results on the *atheism* and *Christianity* subreddits. In general sentiment does not help on this dataset (see row 1). But the NNP and subreddit features again consistently improve recall without hurting precision. And, as above, $\ell_1\ell_2$ regularization shrinks variance (see Figures 4 and 5).

Figure 4 shows histograms of the observed absolute differences between the baseline linear classifier and the proposed augmentations. Adding the proposed features (which capitalize on sentiment and NNP-mentions on specific subreddits) increases absolute median recall by 3.4 percentage points (a relative gain of ∼12%). And this is achieved without sacrificing precision (in contrast to exploiting only sentiment). Furthermore, as we can see in Figures 4 and 5, the proposed regularization strategy shrinks the variance of the classifier. This variance reduction is achieved through greater model sparsity, as can be seen in Figure 6, which improves interpretability. We note that leveraging *only* an ℓ_1 regularization penalty (with the full feature-set) results in very poor performance (median recall and precision of 0.05 and 0.09, respectively). Similarly, the elastic-net strategy [40] (in which we do not specify which features to apply the ℓ_1 penalty to), here achieves a median recall of 0.11 and a median precision of 0.07.

Table 5 reports results on the held-out political test dataset, achieved after training the models on the entirety of the development corpus. To account for the variance inherent to inference via SGD, we performed

	median recall (std. dev.)	median precision (std. dev.)
baseline	0.331 (0.146)	0.148 (0.022)
(overall) sent.	0.351 (0.054)	0.125 (0.003)
NNP	0.364 (0.119)	0.135 (0.021)
NNP $\times$ subreddit	0.357 (0.108)	0.143 (0.020)
NNP+ $\times$ sent. $\times$ subreddit	0.344 (0.116)	0.142 (0.019)
NNP+ $\times$ sent. $\times$ subreddit (ℓ_1 ℓ_2)	0.325 (0.052)	0.141 (0.008)
NNP+ $\times$ sent. $\times$ subreddit + sent.	0.377 (0.104)	0.141 (0.014)
NNP+ $\times$ sent. $\times$ subreddit + sent. (ℓ_1 ℓ_2)	0.370 (0.056)	0.140 (0.008)

Table 5: Results on the held-out political dataset, using the entire development corpus as a training set. Abbreviations are as described in the caption for Figure 4. Due to the variance inherent to the stochastic gradient descent procedure, we repeat the experiment 100 times and report the median performance and standard deviations (of different SGD runs). Results are consistent with those reported for the development corpus.

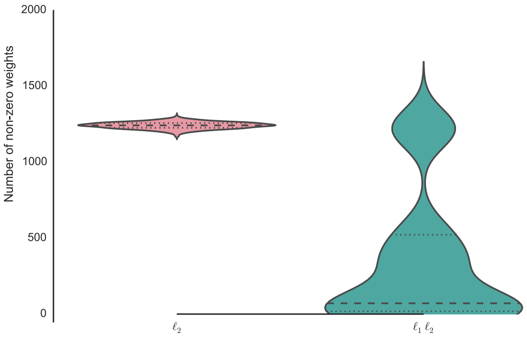


Figure 6: Empirical distributions (violin plots) of non-zero feature counts in the NNP × subreddit model (rows 3 and 4 in Figure 5) using standard ℓ_2-norm (left) and the proposed $\ell_1\ell_2$-norm (right) regularization approaches on the *atheism*/*Christianity* data over 500 independent train/test splits. The composite norm achieves much greater sparsity, resulting in lower variance. This sparsity also (arguably) provides greater interpretability; one can inspect contextual features with non-zero weights.

100 runs of the SGD procedure and report median results from these runs. These results mostly agree with those reported for the development corpus: the proposed strategy improves median recall on the held-out corpus by nearly 4.0 percentage points, at a median cost of about 1 point in precision. By contrast, sentiment alone provides a 2% absolute improvement in recall at the expense of more than 2 points in precision.

To assess the general applicability of the proposed approach, we also evaluate the method on comments from a separate pair of polarized communities: *atheism* and *Christianity*. This dataset was not used during model development. We follow the experimental setup described above.

In this case, capitalizing on the NNP × subreddit features produces a mean 2.3% absolute gain in recall (median: 2.4%) over the baseline approach, with a (very) slight gain in precision. The ℓ_1 ℓ_2 approach achieves a lower expected gain in recall (median: 1.5%), but again shrinks the variance w.r.t. model performance (see Figure 5). Moreover, as we show in Figure 6, this is achieved with a much more compact (sparser) model. We note that for the religion data, inferred sentiment features do not seem to improve performance, in contrast to the results on the political subreddits. At present, we are not sure why this is the case.

These results demonstrate that introducing features that encode entities and user communities (NNPs × subreddit) improve recall for irony detection in comments addressing relatively diverse topics (politics and religion).

We report the interaction features that are the best predictors of verbal irony in the respective subreddits (for both polar community pairs). Specifically, we estimated the weights for every interaction feature using the entire training dataset, and repeated this process 100 times to account for variation due to the SGD procedure.

Table 6 displays the top 10 NNP × subreddit features for the political subreddits, with respect to the mean magnitude of the weights associated with them. We report these means and the standard deviations calculated across the 100 runs. This table implies, for example, that mentions of 'freedom' and 'kenya' indicate irony in the *progressive* subreddit; while mentions of 'obamacare' and 'president' (for example) in the *conservative* subreddit tend to imply irony.

Table 7 reports analagous results for the religion subreddits. Here we can see, e.g., that 'god' is a good predictor of irony in the *atheism* subreddit, and 'professor' is in the *Christianity* subreddit.

We also report the top ranking 'three-way' interaction features that cross NNP's extracted from sentences with subreddits and the inferred sentiment for the political corpus (Table 8). This would imply, e.g., that if a sentence in the *progressive* subreddit conveys an ostensibly positive sentiment about the political commentator 'Ollie',[5] then this sentence is likely to have been intended ironically.

[5] 'Ollie' is a conservative political commentator.

progressive		*conservative*	
feature	weight	feature	weight
freedom	0.102 (0.048)	racist	0.148 (0.043)
god	0.085 (0.045)	news	0.100 (0.044)
christmas	0.081 (0.046)	way	0.078 (0.044)
jesus	0.060 (0.038)	obamacare	0.068 (0.041)
kenya	0.052 (0.035)	white	0.059 (0.037)
brave	0.043 (0.035)	let	0.058 (0.038)
bravo	0.041 (0.035)	course	0.046 (0.033)
know	0.038 (0.030)	huh	0.044 (0.036)
dennis	0.038 (0.029)	education	0.043 (0.032)
ronald	0.036 (0.030)	president	0.039 (0.031)

Table 6: Average weights (and standard deviations calculated across samples) for top 10 NNP × subreddit features from the *progressive* and *conservative* subreddits.

atheism		*Christianity*	
feature	weight	feature	weight
right	0.353 (0.014)	professor	0.297 (0.013)
god	0.324 (0.013)	let	0.084 (0.014)
women	0.214 (0.013)	peter	0.080 (0.019)
christ	0.160 (0.014)	geez	0.054 (0.016)
news	0.146 (0.013)	evil	0.054 (0.015)
trust	0.139 (0.013)	killing	0.053 (0.015)
shit	0.132 (0.015)	liberal	0.049 (0.014)
believe	0.123 (0.013)	antichrist	0.049 (0.014)
great	0.121 (0.016)	rock	0.047 (0.014)
ftfy	0.108 (0.016)	pedophilia	0.046 (0.014)

Table 7: Top 10 NNP × subreddit features from the *atheism* and *Christianity* subreddits (coefficient means and standard deviations).

progressive		*conservative*	
feature	weight	feature	weight
american (+)	0.045 (0.023)	mr (+)	0.041 (0.021)
yay (+)	0.042 (0.022)	cruz (+)	0.040 (0.021)
ollie (+)	0.036 (0.019)	king (+)	0.036 (0.019)
north (+)	0.036 (0.019)	onion (+)	0.035 (0.018)
fuck (+)	0.034 (0.018)	russia (+)	0.034 (0.018)
washington (+)	0.034 (0.018)	oprah (+)	0.030 (0.016)
times* (+)	0.034 (0.018)	science (+)	0.027 (0.015)
world (+)	0.030 (0.016)	math (+)	0.027 (0.015)
magic (+)	0.024 (0.013)	america (+)	0.026 (0.014)
where (+)	0.024 (0.013)	ben (+)	0.020 (0.011)

Table 8: Average weights (and standard deviations) for top 10 NNP × subreddit × sentiment features. The parenthetical '+' indicates that the inferred sentiment was positive. In general, (ostensibly) positive sentiment indicates irony.

Some of these may seem counter-intuitive, such as ostensibly positive sentiment regarding 'Cruz' (as in the conservative senator Ted Cruz) in the conservative subreddit. On inspection of the comments, it would seem Ted Cruz does not find general support even in this community. Example comments include: "Stay classy Ted Cruz" and "Great idea on the talkathon Cruz". The 'mr' and 'king' terms are almost exclusively references to Obama in the *conservative* subreddit. In any case, because these are three-way interaction terms, they are all relatively rare: therefore we would caution against over interpretation here.

The above constitutes one means of exploiting context, at least in linear ('bag-of-words') based models. During the duration of this project, neural networks re-emerged as a dominant NLP technology [**?**], thus bringing to the fore an important question: how can we encode and capitalize on relevant contextual information in neural architectures? We address this in a few ways below.

4 Exploiting Context in Neural Mdoels

In this section we describe our progress on designing neural models [9, **?**] for verbal irony detection. Such models have quite recently become enormously popular for NLP recently due to their exceptional performance.[6] Neural models leverage distributed 'embeddings' of words [19], which provide a richer amount of context than unstructured, 'bag-of-words' (BoW) representations.

In recent work, we have found that Convolutional Neural Networks (CNNs) in particular perform quite well for text classification [39], besting 'linear' models based on BoW representations. Thus in this section we describe our progress on innovative approaches we have developed for classifying short pieces of social media text (forum posts and tweets) as having been intended ironically or not. For this project we have made a few key innovations that allow CNNs to exploit additional contextual information.

4.1 Multi-Group Norm Constraint CNN (MGNC-CNN)

In this subsection we introduce a novel, simple Convolution Neural Network (CNN) architecture – multi-group norm constraint CNN (MGNC-CNN) – that capitalizes on multiple sets of word embeddings for sentence classification, thus capturing additional (linguistic/semantic) context. MGNC-CNN extracts features from input embedding sets independently and then joins these at the penultimate layer in the network to form a final feature vector. We then adopt a group regularization strategy that differentially penalizes weights associated with the subcomponents generated from the respective embedding sets. This model is much simpler than comparable alternative architectures and requires substantially less training time. Furthermore, it is flexible in that it does not require input word embeddings to be of the same dimensionality. Here we show that MGNC-CNN consistently outperforms baseline models with respect to classifying posts on the reddit corpus described as sarcastic or not.

4.1.1 The Model

We first review standard one-layer CNN (which exploits a single set of embeddings) for sentence classification [14], and then propose our augmentations, which exploit multiple embedding sets.

Basic CNN

In this model we first replace each word in a sentence with its vector representation, resulting in a sentence matrix $\mathbf{A} \in R^{sxd}$ where s is the (zero-padded) sentence length, and d is the dimensionality of the embeddings. We apply a convolution operation between linear filters with parameters $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_k$ and the sentence matrix. For each $\mathbf{w}_i \in R^{hxd}$, where h denotes 'height', we slide filter i across $\mathbf{A}$, considering 'local regions' of h adjacent rows at a time. At each local region, we perform element-wise multiplication and then take the element-wise sum between the filter and the (flattened) sub-matrix of $\mathbf{A}$, producing a scalar. We do this for each sub-region of $\mathbf{A}$ that the filter spans, resulting in a feature map vector $\mathbf{c}_i \in R^{(sh+1)x1}$.

[6] The resurgence of neural models can be attributed to a number of factors, including more available memory, faster Graphical Processing Units (GPUs), and better algorithms for parameter estimation.

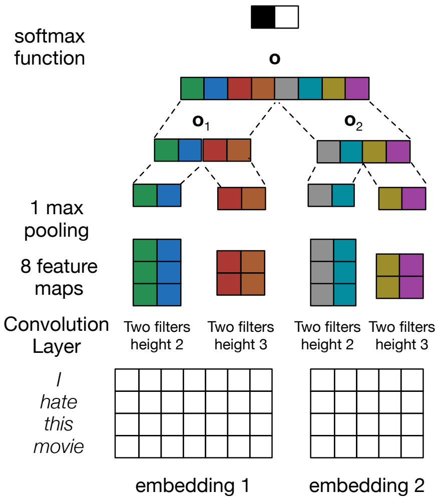


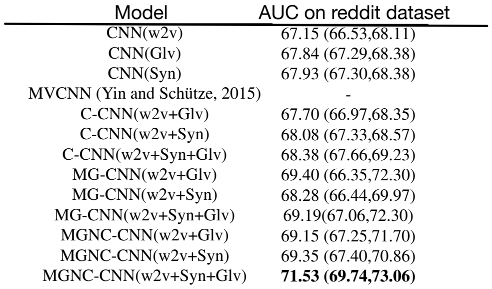
Figure 7: Illustration of MG-CNN/MGNC-CNN. The filters applied to the respective embeddings are completely independent. MG-CNN applies a max norm constraint to $\mathbf{o}$, while MGNC-CNN applies max norm constraints on $\mathbf{o}_1$ and $\mathbf{o}_2$ independently (group regularization). Note that one may easily extend the approach to handle more than two embeddings at once.

We can use multiple filter sizes with different heights, and for each filter size we can have multiple filters. Thus the model comprises k weight vectors $\mathbf{w}_1, \mathbf{w}_2, ..., \mathbf{w}_k$, each of which is associated with an instantiation of a specific filter size. These in turn generate corresponding feature maps $\mathbf{c}_1, \mathbf{c}_2, ...\mathbf{c}_k$ with dimensions varying with filter size. A 1-max pooling operation is applied to each feature map, extracting the largest number $\mathbf{o}_i$ from each feature map i. Finally, we combine all $\mathbf{o}_i$ together to form a feature vector $\mathbf{o} \in R^k$ to be fed through a softmax function for classification. We regularize weights at this level in two ways. (1) Dropout, in which we randomly set elements in o to zero during the training phase with probability p, and multiply p with the parameters trained in $\mathbf{o}$ at test time. (2) An $\ell 2$ norm penalty, for which we set a threshold λ for the $\ell 2$ norm of $\mathbf{o}$ during training; if this is exceeded, we rescale the vector accordingly. For more details, see [39].

This variant of CNN can exploit only one notion of a word's semantics, i.e., that encoded in the input matrix. However, the meaning of words is diverse and context-dependent. This is especially important in the context of verbal irony; it is critical to capture alternative potential meanings of words. To realize this aim, we have developed the Multi-Group Norm Constraint CNN (MGNC-CNN), which jointly exploits multiple sets of word embeddings, i.e., multiple sets of word semantics.

MG-CNN

Assuming we have m word embeddings with corresponding dimensions d_1, d_2, ... d_m; we can simply treat each word embedding independently. In this case, the input to the CNN comprises multiple sentence matrices $\mathbf{A}_1$, $\mathbf{A}_2$, ... $\mathbf{A}_m$, where each $\mathbf{A}_l \in R^{sxd_l}$ may have its own width d_l. We then apply different groups of filters $\mathbf{w}_1$, $\mathbf{w}_2$, ...$\mathbf{w}_m$ independently to each $\mathbf{A}_l$, where $\mathbf{w}_l$ denotes the set of filters for $\mathbf{A}_l$. As in basic CNN, $\mathbf{w}_l$ may have multiple filter sizes, and multiple filters of each size may be introduced. At the classification layer we then obtain a feature vector $\mathbf{o}_l$ for each embedding set, and we can simply concatenate these together to form the final feature vector $\mathbf{o}$ to feed into the softmax function, where $\mathbf{o} = \mathbf{o}_1 \oplus \mathbf{o}_2 ... \oplus \mathbf{o}_m$. This representation contains feature vectors generated from all sets of embeddings under consideration. We call this method multiple group CNN (MG-CNN). Here groups refer to the features generated from different embeddings. Note that this differs from 'multi-channel' models because at the convolution layer we use different filters on each word embedding matrix independently, whereas in a standard multi-channel approach each filter would consider all channels simultaneously and generate a scalar from all channels at each local region. As above,



Model	AUC on reddit dataset
CNN(w2v)	67.15 (66.53,68.11)
CNN(Glv)	67.84 (67.29,68.38)
CNN(Syn)	67.93 (67.30,68.38)
MVCNN (Yin and Schütze, 2015)	-
C-CNN(w2v+Glv)	67.70 (66.97,68.35)
C-CNN(w2v+Syn)	68.08 (67.33,68.57)
C-CNN(w2v+Syn+Glv)	68.38 (67.66,69.23)
MG-CNN(w2v+Glv)	69.40 (66.35,72.30)
MG-CNN(w2v+Syn)	68.28 (66.44,69.97)
MG-CNN(w2v+Syn+Glv)	69.19(67.06,72.30)
MGNC-CNN(w2v+Glv)	69.15 (67.25,71.70)
MGNC-CNN(w2v+Syn)	69.35 (67.40,70.86)
MGNC-CNN(w2v+Syn+Glv)	**71.53 (69.74,73.06)**

Figure 8: Results (AUC) of variants of MGNC-CNN v baseline approaches on the reddit corpus. MGNC-CNN with three word embeddings performs best here.

we impose a max ℓ_2 norm constraint on the final feature vector $\mathbf{o}$ for regularization.

MGNC-CNN

We propose an augmentation of MG-CNN, Multi-Group Norm Constraint CNN (MGNC- CNN), which differs in its regularization strategy. Specifically, in this variant we impose grouped regularization constraints, independently regularizing subcomponents $\mathbf{o}_l$ derived from the respective embeddings, i.e., we impose separate max norm constraints λ_l for each $\mathbf{o}_l$ (where l again indexes embedding sets); these λ_l hyper-parameters are to be tuned on a validation set. Intuitively, this method aims to better capitalize on features derived from word embeddings that capture discriminative properties of text for the task at hand by penalizing larger weight estimates for features derived from less discriminative embeddings. See Figure 7.

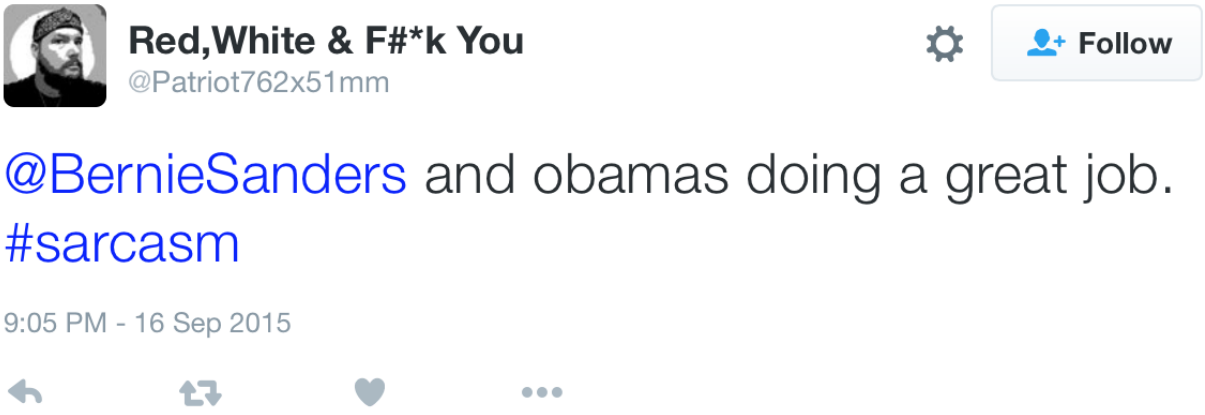
4.1.2 Experimental Setup and Results

We consider three sets of word embeddings for our experiments: (i) word2vec [16] is trained on 100 billion tokens of Google News dataset; (ii) GloVe [23] is trained on aggregated global word-word co-occurrence statistics from Common Crawl (840B tokens); and (iii) syntactic word embedding trained on dependency-parsed corpora. These capture different kinds of context. These three embedding sets happen to all be 300-dimensional, but our model could accommodate arbitrary and variable sizes.

We compared our proposed approaches to a standard CNN that exploits a single set of word embeddings [14]. We also compared to a baseline of simply concatenating embeddings for each word to form long vector inputs. We refer to this as Concatenation-CNN C-CNN. For all multiple embedding approaches (C-CNN, MG-CNN and MGNC-CNN), we explored two combined sets of embedding: word2vec+Glove, and word2vec+syntactic, and one three sets of embedding: word2vec+Glove+syntactic. For all models, we tuned the $\ell 2$ norm constraint λ over the range $\{1, 1, 3, 9, 81, 243\}$ on a validation set. For instantiations of MGNC-CNN in which we exploited two embeddings, we tuned both λ_1 and λ_2; where we used three embedding sets, we tuned λ_1, λ_2 and λ_3.

We performed 10-fold cross validation, creating nested development sets with which to tune hyperparameters. For all experiments we used filters sizes of 3, 4 and 5 and we created 100 feature maps for each filter size. We applied 1 max-pooling and dropout (rate: 0.5) at the classification layer. For training we used back-propagation in mini-batches and used AdaDelta as the stochastic gradient descent (SGD) update rule, and set mini-batch size as 50. In this work, we treat word embeddings as part of the parameters of the model, and update them as well during training. We only tuned the max norm constraint(s), fixing all other hyperparameters

In Figure 4.1.2 we report results on the irony corpus (described above) in terms of the area under the sensitivity/specificity curve, which measures overall discriminative performance – this is more appropriate than, say, accuracy here because the dataset is very imbalanced (most comments are unironic). Our model with three embeddings achieves a 4 point gain in AUC compared to the baseline neural model, demonstrating



Red,White & F#*k You
@Patriot762x51mm

@BernieSanders and obamas doing a great job. #sarcasm

9:05 PM - 16 Sep 2015

Figure 9: An illustrative tweet.

the benefits of capturing more context using our approach. Note that MGNC-CNN provides a substantial boost over baselines with respect to discerning verbal irony in reddit posts.

MGNC-CNN, just reviewed, captures additional lexical context; but irony depends on the speaker. We therefore now shift our focus to a model that attempts to encode information about speakers (users) to inform predictions.

4.2 Modelling Context with User Embeddings

Existing social media analysis systems are hampered by their inability to accurately detect and interpret figurative language. This is particularly relevant in domains like the social sciences and politics, in which the use of figurative communication devices such as verbal irony (roughly, sarcasm) is common. As we have emphasized throughout this project, sarcasm is often used by individuals to express opinions on complex matters and regarding specific targets [33, 6].

Early computational models for verbal irony and sarcasm detection tended to rely on shallow methods that exploited conditional token count regularities. But, as we argued above, lexical clues alone are insufficient to discern ironic intent. Appreciating the *context* of utterances is critical for this; even for humans [36]. Consider the sarcastic tweet in Figure 9 (ignoring for the moment the attached `#sarcasm` hashtag). Without knowing the author's political leanings, it would be difficult to conclude with certainty whether the remark was intended sarcastically or in earnest.

Above (Section 3.4) we developed a method that capitalizes on contextualizing information to inform predictions [35]; others have recently proposed similar approaches [2]. However, these approaches require the design and implementation of complex features that explicitly encode the content and (relevant) context of messages to be classified. This feature engineering is labor intensive, and depends on the specifics of the dataset at hand (i.e., particular social media platform), external tools and resources. Therefore, deploying such systems in practice is expensive, time-consuming and unwieldy. Here we propose a novel approach to sarcasm detection on social media that does not require extensive manual feature engineering but still captures the relevant context. Specifically, we develop a neural model that learns to represent and exploit embeddings of both *content* and *context*. For the former, we induce vector lexical representations via a convolutional layer (as above, in Section 4.1). For the latter, our model learns *user embeddings*. Inference concerning whether an utterance (tweet) was intended ironically (or not) is then modelled as a joint function of lexical representations and corresponding author embeddings.

4.3 Learning User Embeddings

Our goal is to learn representations (vectors) that encode latent aspects of users and capture homophily by projecting similar users into nearby regions of the embedding space. Our hypothesis is that such representations will naturally capture some of the signals that have been described in the literature as important indicators of sarcasm, for example contrasts between what someone believes and what they have ostensibly expressed [5].

To estimate user embeddings we adopt an approach similar to that described in the preliminary work of [17]. In particular, we capture relations between users and the content they produce by optimizing the conditional probability of texts, given their authors (or, more precisely, given the vector representations of their authors). This method is akin to [16]'s *Paragraph Vector* model, which jointly estimates embeddings for words and paragraphs by learning to predict the occurrence of a word w within a paragraph p conditioned on the (learned) representation for p.

Given a sentence $S = \{w_1, \ldots, w_N\}$ where w_i denotes a word drawn from a vocabulary $\mathcal{V}$, we aim to maximize the following probability:

$$\begin{aligned} P(S|\text{user}_j) = & \sum_{w_i \in S} \log P(w_i|\mathbf{u}_j) \\ & + \sum_{w_i \in S} \sum_{w_k \in C(w_i)} \log P(w_i|\mathbf{e}_k) \end{aligned} \tag{8}$$

where $C(w_i)$ denotes the set of words in a pre-specified window around word w_i, $\mathbf{e}_k \in \mathbb{R}^d$ and $\mathbf{u}_j \in \mathbb{R}^d$ denote the embeddings of word k and user j, respectively. This objective function encodes the notion that the occurrence of a word w, depends both on the author of S and it's neighbouring words.

The conditional probabilities in Equation 8 can be estimated with log-linear models of the form:

$$P(w_i|\mathbf{x}) = \frac{\exp(\mathbf{W}_i \cdot \mathbf{x} + \mathbf{b}_i)}{\sum_{k=1}^{Y} \exp(\mathbf{W}_k \cdot \mathbf{x} + \mathbf{b}_k)} \tag{9}$$

Where $\mathbf{x}$ denotes a feature vector, $\mathbf{W}_k$ and $\mathbf{b}_k$ are the weight vectors and bias for class k. In our case, we treat words as classes to be predicted. Calculating the denominator thus requires summing over all of the words in the (large) vocabulary, an expensive operation. To avoid this computational bottleneck, we approximate the term $P(w_i|\mathbf{e}_k)$ with Hierarchical Softmax [21].[7]

Our primary goal is to learn meaningful user representations. In particular, we seek representations that are predictive of individual word-usage patterns. In light of this motivation, we approximate $P(w_i|\mathbf{u}_j)$ via the following hinge-loss objective which we aim to minimize:

$$\begin{aligned} & \mathcal{L}(w_i, \text{user}_j) = \\ & \qquad \sum_{w_l \in V, w_l \notin S} \max(0, 1 - \mathbf{e}_i \cdot \mathbf{u_j} + \mathbf{e}_l \cdot \mathbf{u_j}) \end{aligned} \tag{10}$$

Where each w_l (and corresponding embedding, $\mathbf{e}_l$) is a *negative example*, i.e., a word not in the sentence under consideration, which was authored by user j. The intuition is that in the aggregate, such words are less likely to be employed by user j than words observed in sentences she has authored. Thus minimizing this objective attempts to induce a representation that is discriminative with respect to word usage.

In practice, V will be very large and hence we approximate the objective via *negative sampling*, a variant of Noise Contrastive Estimation.[8] The idea is to approximate the objective function in a binary classification task by learning to discriminate between observed positive examples (sampled from the true distribution)

[7] As implemented in `Gensim` [24].

[8] See [11] for notes on Negative Sampling and Noise Contrastive Estimation

and *pseudo*-negative examples (sampled from a large space of predominantly negative instances). Intuitively, this shifts probability mass to plausible observations.

This approach works well in representation learning tasks when a sufficient amount of training data is available [9]. However, we have access to only a limited amount of text for each user (see Section 4.4). We hypothesize that this problem can be alleviated by carefully selecting the negative samples that mostly contribute to "push" the vectors into the appropriate region of the embedding space (i.e., closer to the words commonly employed by a given user and far from other words). This of course requires designing a strategy for selectively sampling negative examples. One straightforward approach would be to sample from a user-specific unigram model, informing which words are less likely to be used by that user. But estimating the parameters of such model with scarce data would be prone to overfitting. Instead, we sample from a unigram distribution estimated with maximum likelihood from all the data. The intuition here is that the representations should be discriminative of the most distinct traits of the specific user.

4.3.1 Proposed Model

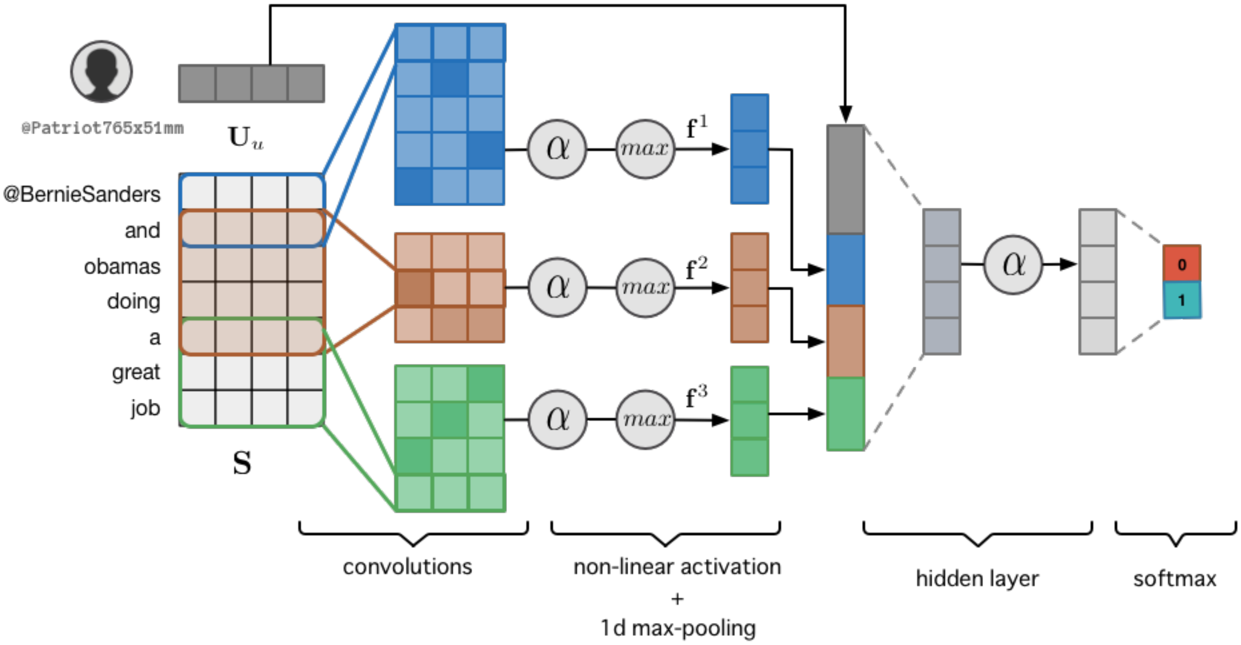


Figure 10: Illustration of our deep neural network for sarcasm detection. The model learns to represent and exploit embeddings of both *content* and *users* in social media.

We now present the details of our proposed novel sarcasm detection model. Given a message s authored by user u, we wish to capture both the relevant aspects of the *content* and the relevant *contextual* information about the author. To represent the content, we use pre-trained word embeddings as the input to a convolutional layer that extracts high-level features. More formally, let $\mathbf{E} \in \mathbb{R}^{d \times |\mathcal{V}|}$ be a pre-trained word embedding matrix, where each column represents a word from the vocabulary $\mathcal{V}$ as a d dimensional vector. By selecting the columns of $\mathbf{E}$ corresponding to the words in S, we form the sentence matrix:

$$\mathbf{S} = \begin{bmatrix} \mathbf{e}_1 \\ \vdots \\ \mathbf{e}_m \end{bmatrix} \tag{11}$$

A convolutional layer is composed of a set of filters $\mathbf{F} \in \mathbb{R}^{d \times h}$ where h is the *height* of the filter. Filters *slide* across the input, extracting h-gram features that constitute a feature map $\mathbf{m} \in \mathbb{R}^{|S|-h+1}$, where each entry is obtained as

$$\mathbf{m}_i = \alpha(\mathbf{F} \cdot \mathbf{S}_{[i:i-h+1]} + b) \tag{12}$$

with $i = 1, \ldots i - h + 1$. Here, $\mathbf{S}_{[i:j]}$ denotes a sub-matrix of $\mathbf{S}$ (from row i to row j), $b \in \mathbb{R}$ is an additive bias and $\alpha(\cdot)$ denotes a non-linear activation function, applied element-wise. We transform the resultant feature map into a scalar using *max-pooling*, i.e., we extract the largest value in the map. We use 3 filters (with varying heights) each of which generates M feature maps that are reduced to a vector $\mathbf{f}^k = [max(\mathbf{m}^1) \ \oplus max(\mathbf{m}^2) \ldots \oplus \ max(\mathbf{m}^M)]$, where $\oplus$ denotes concatenation. We set $\alpha(\cdot)$ to be the *Rectified Linear Unit* activation function [22]. The output of all the filters is then combined to form the final representation $\mathbf{c} = [\mathbf{f}^1 \oplus \mathbf{f}^2 \oplus \mathbf{f}^3]$. We will denote this feature vector of a specific sentence s by $\mathbf{c}_s$.

To represent the context, we assume there is a user embedding matrix $\mathbf{U} \in \mathbb{R}^{d \times N}$, where each column represents one of N users by a d dimensional embedding. The parameters of this embedding matrix can be initialized randomly or using the user embedding estimation approach described above. We simply map the author of the message into the user embedding space by selecting the corresponding column of $\mathbf{U}$. Letting $\mathbf{U}_u$ denote the user embedding of author u, we formulate our sarcasm detection model as follows:

$$\begin{gathered} P(y = k|s, u; \theta) \propto \mathbf{Y}_k \cdot g(\mathbf{c}_s \oplus \mathbf{U}_u) + \mathbf{b}_k \\ g(\mathbf{x}) = \alpha(\mathbf{H} \cdot \mathbf{x} + \mathbf{h}) \end{gathered} \tag{13}$$

where $g(\cdot)$ denote the activations of a hidden layer, capturing the relations between the content and context representations, and $\theta = \{\mathbf{Y}, \mathbf{b}, \mathbf{H}, \mathbf{h}, \mathbf{F}^1, \mathbf{F}^2, \mathbf{F}^3, \mathbf{E}, \mathbf{U}\}$ are parameters to be estimated during training. Here, $\mathbf{Y} \in \mathbb{R}^{2 \times z}$ and $\mathbf{b} \in \mathbb{R}^2$ are the weights and bias of the output layer; $\mathbf{H} \in \mathbb{R}^{z \times 3M+d}$ and $\mathbf{h} \in \mathbb{R}^z$ are the weights and bias of the hidden layer; and $\mathbf{F}^i$ are the weights of the convolutional filters. Figure 10 provides a schematic depicting our model.

4.4 Experimental Setup

We replicated the experimental setup used by Bamman et al. [2] using (a subset of) the same Twitter corpus. Labels were inferred from self-declarations of sarcasm, i.e., a tweet is considered sarcastic if it contains the hashtag `#sarcasm` or `#sarcastic` and deemed non-sarcastic otherwise.[9] For each author and mentioned user, we scraped additional tweets from their Twitter feed. Due to restrictions in the Twitter API, we were only able to crawl at most 1000 historical tweets per user.[10] Furthermore, we were unable to collect historical tweets for a significant proportion of the users, thus, we discarded messages for which no contextual information was available, resulting in a corpus of $11,541$ tweets involving $12,500$ unique users (authors and mentioned users). It should also be noted that our historical tweets were posted *after* the ones in the corpus used for the experiments.

4.5 Baselines

We reimplemented [2]'s sarcasm detection model. This a simple, logistic-regression based classifier that exploits rich feature sets to achieve strong performance. These are detailed at length in [2], but we summarize briefly here:

- **tweet-features**: These encode attributes of the target tweet text. Specifically, this includes: uni- and bi-gram bag of words (BoW) features; Brown cluster [4] indicators; unlabeled dependency bigrams (both BoW and with Brown cluster representations); part-of-speech features (inferred automatically with an off the shelf model); spelling and abbreviation features; inferred sentiment, at both the tweet and word level; and 'intensifier' indicators.

[9] Note that this is a form of noisy supervision, as of course all sarcastic tweets will not be explicitly flagged as such.

[10] The original study [2] was done with at most $3,200$ historical tweets.

- **author-features**: These features aim to encode attributes of the author. These include: historically 'salient' terms used by the author; the inferred distribution over topics (from Latent Dirichlet Allocation [3]) historically tweeted about by the user; inferred sentiment historically expressed by the user; and author profile information (e.g., profile BoW features).

- **audience-features**: These capture properties of the *addressee* of tweets (i.e., the person to whom the author is tweeting), in those cases that a tweet is directed at someone (via the @ symbol). A subset of these duplicate the author features for the addressee: historical topics, historical salient terms, profile information, and profile BoW information is encoded. Additionally, author/audience interaction features are introduced, which capture similarity between the author and addressee, w.r.t. inferred topic distributions. Finally, this set includes a feature capturing the frequency of past communication between the author and addressee.

- **response-features**: For 'response tweets', i.e., those written in response to another tweet, this set of features captures information relating the two. This includes BoW features of the original tweet and pairwise Brown cluster indicator features, which encode Brown clusters observed in both the original and response tweet.

We emphasize that implementing this rich set of features took considerable time and effort. This motivates our approach, which aims to effectively induce and exploit contextually-aware representations without manual feature engineering.

To assess the importance of modelling contextual information for sarcasm detection, we considered two groups of models as baselines: the first only takes into account the content contained in the target tweet. The second combines lexical clues with contextual information. The first group includes the following models:

- UNIGRAMS: ℓ_2-regularized logistic regression classifier with binary unigrams as features.

- TWEET ONLY: ℓ_2-regularized logistic regression classifier with binary unigrams and **tweet-features**.

- NBOW: Logistic regression with neural word embeddings as features. Given a sentence matrix $\mathbf{S}$ (Eq. 11) as input, a d-dimensional feature vector is computed by summing the individual word embeddings.

- NLSE: The Non-linear subspace embedding model due to [1]. This approach consists of adapting embeddings for a specific task, by learning a projection into a small subspace that captures the most relevant latent aspects encoded by the pre-trained embeddings. Given a sentence matrix $\mathbf{S}$, each word vector is first projected into the subspace and then transformed through an element-wise sigmoid function. The final sentence representation is obtained by summing the (adapted) word embeddings and passed into a softmax layer that outputs the predictions.

- CNN: A simplified version of the model presented Section 4.3.1, using only features extracted from the convolutional layer acting on the lexical content. This is the CNN model for text classification proposed by Kim [14].

The second group of baselines consists of the following models:

- TWEET+*: ℓ_2-regularized logistic regression classifier with a combination of **tweet-features** and the each of the aforementioned [2] feature sets.

- CNN+CONTEXT: A simplified version of our neural model for sarcasm detection, without the hidden layer and with the user embeddings initialized at random.

- CNN+USER2VEC: The same model as above, but initializing the user embeddings with the approach described in Section 4.3.

- DeepCNN+Context: Our neural model for sarcasm detection with the user embeddings initialized at random.
- DeepCNN+user2vec-*: Our neural model for sarcasm detection utilizing pre-trained user embeddings. We compared different approaches for the negative sampling procedure, namely, sampling from a unigram distribution (user2vec-BackDist) and sampling uniformly at random from the vocabulary (user2vec-UnifRand).

4.5.1 Pre-Training Word and User Embeddings

We first trained [19]'s *skip-gram* model variant to induce word embeddings using the union of a dataset of 52 Million unlabeled tweets and the dataset to be used in the experiments. The latter includes 5 Million historical tweets collected from users.

To induce user embeddings, we estimated a unigram distribution using a maximum likelihood estimate. Then, for each word in a tweet, we extracted 15 negative samples (for the first term in Eq.8) and used the skip-gram model to pre-compute the conditional probabilities of words occurring in a window of size 5 (for the second term in Eq.8). Finally, Equation 8 was minimized via Stochastic Gradient Descent on 90% of the historical data, holding out the remainder for validation and using the $P(\text{tweet text}|\text{user})$ as early stopping criteria.

Note that the parameters for each user only depend on their own tweets; this allowed us perform these computations in parallel to speed-up the training.

4.5.2 Model Training and Evaluation

Evaluation was performed via 10-fold cross-validation. For each split, we fit models to 80% of the data, tuned them on 10% and tested on the remaining, held-out 10%. These data splits were kept fixed in all the experiments. For the linear classifiers, in each split, the regularization constant was selected with a grid search over the range $C = [1e^{-4}, 1e^{-3}, 1e^{-2}, 1e^{-1}, 1, 10]$ using the training set to fit the model and evaluating on the tuning set. After selecting the best regularization constant, the model was re-trained on the union of the train and tune sets, and evaluated on the test set.

To train our neural model, we first had to choose a suitable architecture and hyperparameter set. However, selecting the optimal network parametrization would require an extensive search over a large configuration space. Therefore, in these experiments, we followed the recommendations in [39], focusing our search over combinations of filter heights $H = [(1, 3, 5), (2, 4, 6), (3, 5, 7), (4, 6, 8), (5, 7, 9)]$, number of feature maps $K = [100, 200, 400, 600]$, size of the hidden layer $Z = [25, 50, 75, 100]$ and dropout rates $D = [0.0, 0.1, 0.3, 0.5]$.

We performed random search by sampling without replacement over half of the possible configurations. For each data split, 20% of the training set was reserved for early stopping. We compared the sampled configurations by fitting the model on the remaining training data and testing on the tune set. After choosing the best configuration, we re-trained the model on the union of the train and tune set (again reserving 20% of the data for early stopping) and evaluated on the test set.

We trained the model by minimizing the cross-entropy error between the predictions and true labels, the gradients w.r.t to the network parameters were computed with backpropagation [26] and model weights were updated with the AdaDelta rule [37].

4.5.3 Results

Classification Results

Figure 4.5.3 presents the main experimental results. In Figure 11, we show the performance of linear classifiers with the manually engineered feature sets proposed by Bamman et al. [2]. Our results differ slightly from those originally reported. Nonetheless, we observe the same general trends: namely, that including contextual features significantly improves the performance, and that the biggest gains are attributable to features encoding information about the authors of tweets.

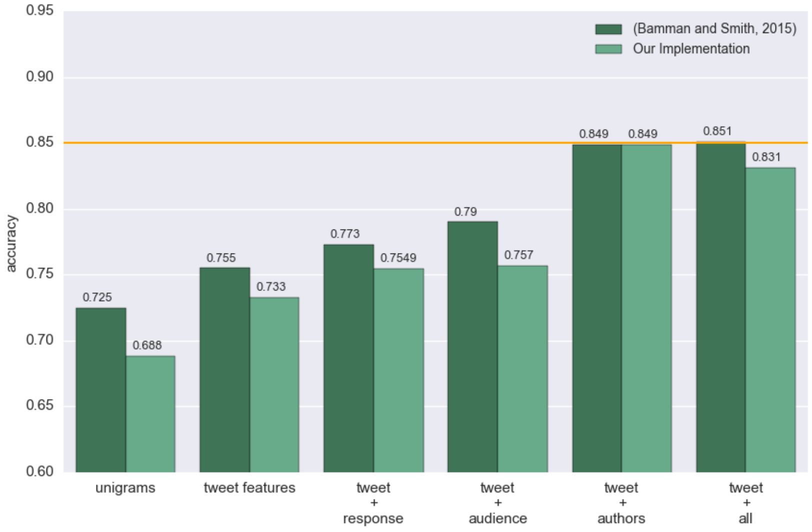


Figure 11: Performance of the linear classifier baselines. We include the results reported by [2] as a reference. Discrepancies between their reported results and those we achieved with our re-implementation reflect the fact that their experiments were performed using a significantly larger training set and more historical tweets than we had access to.

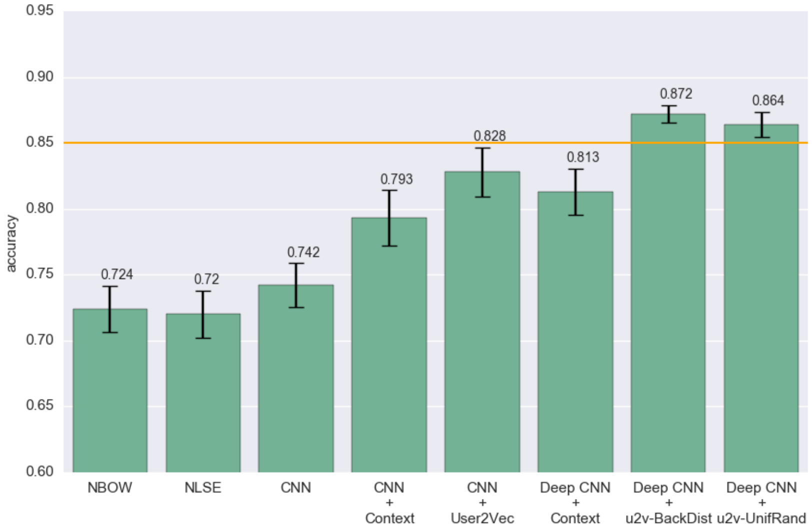


Figure 12: Performance of the proposed neural models. We compare simple neural models that only consider the lexical content of a message (first 3 bars) with architectures that explicitly model the context. Bars 4 and 5 show the gains obtained by pre-training the user embeddings. The last 2 bars compare different negative sampling procedures for the user embedding pre-training. The horizontal line corresponds to the best performance achieved via linear models with rich feature sets. Performance was measured in terms of average accuracy over 10-fold cross-validation; error bars depict the variance.

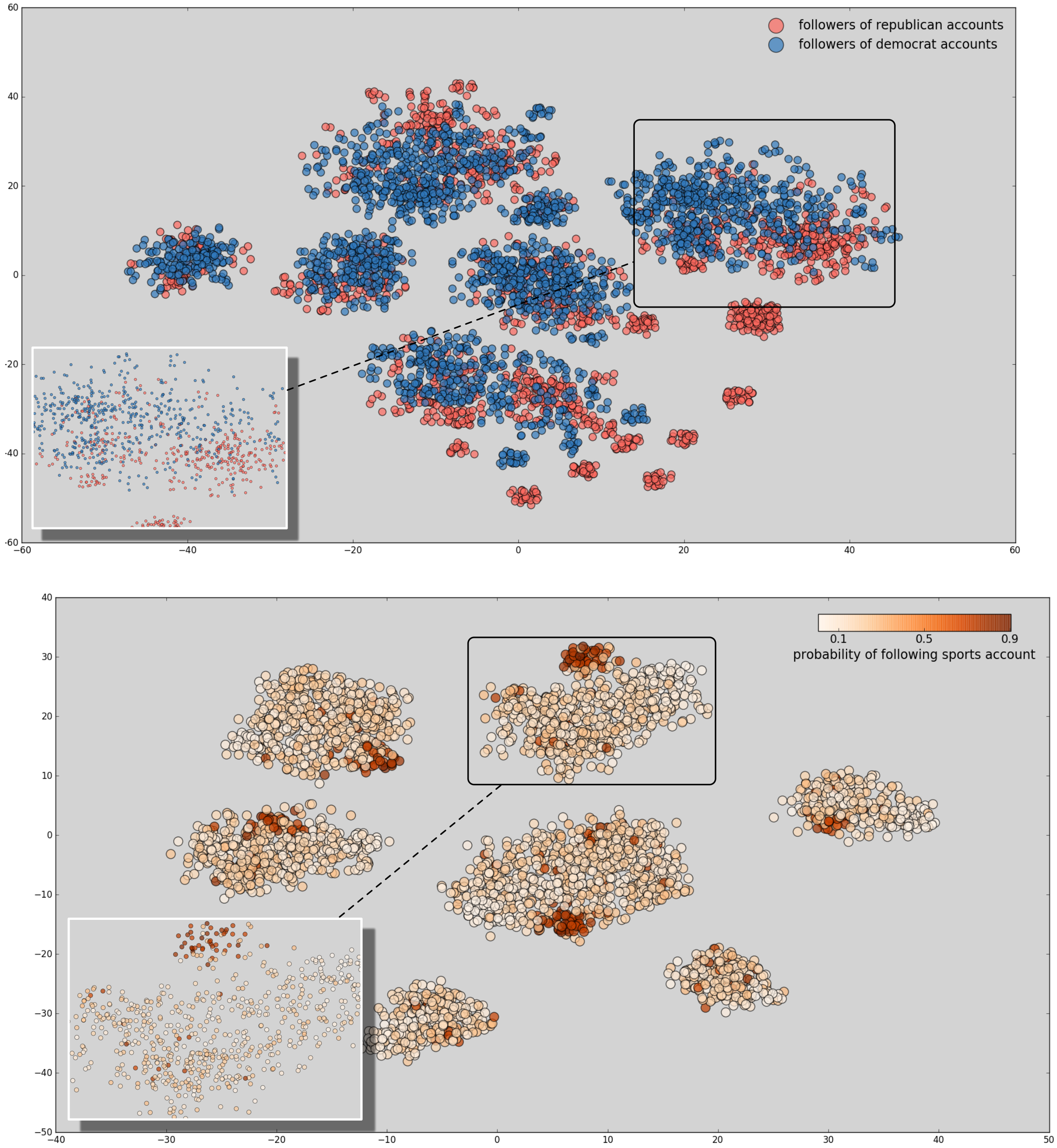


Figure 13: T-SNE projection of the user embeddings into 2-dimensions. The users are color coded according to their political preferences and interest in sports. The visualization suggests that the learned embeddings capture some notion of homophily. Top: Users colored according to the politicians they follow on Twitter: the blue circles represent users that follow at least one of the (democrats) accounts: *@BarackObama*, *@HillaryClinton* and *@BernieSanders*; the red triangles represent users that follow at least one of the (republicans) accounts: *@marcorubio*, *@tedcruz* and *@realDonaldTrump*. Users that follow accounts from both groups were excluded. We can see that users with a similar political leaning tend to have similar vectors. Bottom: Users colored with respect to the likelihood of following a sports account. The 500 most popular accounts (according to the authors in our training data) were manually inspected and 100 sports related accounts were selected, e.g., *@SkySports*, *@NBA* and *@cristiano*. We should note that users for which the probabilities lied in the range between $0.3 - 0.7$ were discarded to emphasize the extremes.

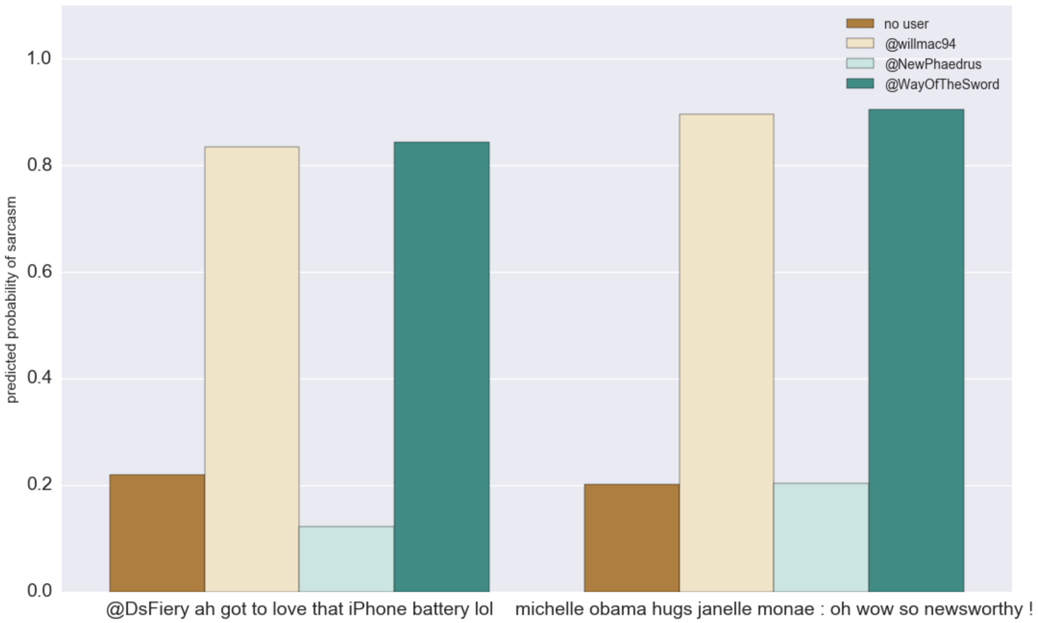


Figure 14: Effect of different user representations in the prediction of sarcasm.

The results of the proposed neural model variants are shown in Figure 12. Once again, we find that modelling the context (i.e., the author) of a tweet yields significant gains in accuracy. The difference is that here the network jointly *learns* appropriate user representations, lexical feature extractors and, finally, the classification model.

We observed that improvements are realized by pre-training the user embeddings (we elaborate on this in the following subsection). We see further improvements when we introduce an additional hidden layer that captures the *interactions* between the context (i.e., user vectors) and the content (lexical vectors). This is intuitively agreeable: the recognition of sarcasm is possible when we jointly consider the speaker and the utterance at hand.

We also compared the effect of obtaining negative samples uniformly at random with sampling from a unigram distribution. The experimental results show that the latter improves the accuracy of the model by 0.8%. We believe the reason is that using the most likely words (under the model) as negative samples helps by pushing the user vectors away from non-informative words and simultaneously closer to the most discriminative words for that user.

User Embedding Analysis

We now investigate the user embeddings in more detail. In particular, we are interested in two questions: first, what aspects are being captured in these representations; and second, how they contribute to the improved performance in our model. In Figure 13, we plot a projection of the high-dimensional vector space where the users are represented into two-dimensions. This visualization suggests that the learned user embeddings are able to uncover latent aspects such as political preferences and interests (e.g., sports). Moreover, the embeddings seem to capture a notion of homophily, i.e. similar users have similar vector representations. In Figure 14, we present two examples that were misclassifed by a simple CNN along with the predicted probabilities of being a sarcastic post. We also show how these predicted probabilities change if we include contextual information.

In this section, we have introduced a novel, neural model for automatically recognizing sarcastic utterances on social media (in this case, Twitter). Our model jointly exploits representations learned for users and tweets, thus integrating information about the speaker and what she has said. This is accomplished without manual feature engineering. Nonetheless, our model *outperforms* (by over 2% in absolute accuracy) a recently proposed baseline model [2] that exploits an extensive, hand-crafted set of features encoding user attributes and other contextual information. And it is more general than the context-aware linear-model we introduced above, which relied on particular aspects of the reddit corpus as contextualizing information (i.e., the subreddit structure).

5 Toward a new conceptual framework

In the past year, PI Beaver has been developing a general framework on sociolinguistic signaling that we believe can be used to contextualize and inform formal models of verbal irony moving forwards. He has presented preliminary results in multiple recent colloquia (at: Yale University, the University of Chicago, and the Leibniz-Zentrum Allgemeine Sprachwissenschaft in Berlin; and he will present the full model in a jointly authored book "Politics of Language" under contract with Princeton University Press, ms. to be delivered to publisher 1/1/2018). Here we summarize in detail how this conceptual model applies within the present project, and suggests novel ways of moving forward.

Irony is a mechanism that can allow contentious opinions to be conveyed, preserving plausible deniability, in that speakers avoid publicly committing to their true belief, thus leaving their underlying attitude less than completely overt. We thus predict that levels of irony will be related to the contentiousness of the issue, and the social cost of making a public commitment. The best performing models we have built within the project depend not only on the raw text of the utterance being analyzed, but on group membership. It is important to recognize that the relevance of social grouping for irony detection is not simply a random feature that we have used opportunistically, since irony has social significance for those groupings.

Notably, irony serves gate-keeper functions within social groups. First, irony is hard to grasp without high context, often by design, and thus solidifies the bonds between those who share sufficient context to recognize it. This is a fundamental issue for intelligence-related or other irony detection procedures, because the premise of automated approaches is that the high context requirements can be overcome through use of sophisticated socially aware modeling. A second gate-keeper function, whereby irony both signals social group membership and acts to exclude non-members, is achieved because the use of irony indicates that an opposing view is not merely incorrect, but should not even be taken seriously, thus showing extreme disrespect both toward opposing views and towards the out-groups that hold those views. Since those with opposing views are not being taken seriously, this can in turn indicate that they are not welcome in a conversation or a community, and the extreme diminishment of an opposing view can prevent emotional obstacles to those sympathetic to those views who would otherwise wish to remain within the conversation. An extreme version of such a gate-keeper function is found in the so-called "weaponized irony" of the contemporary alt-right movement, in which such a high level of use of irony is maintained that there is no clear public record of what exactly is intended, and this in turn is used as a mask for performing transgressive speech acts (e.g. holocaust denial) of such extreme emotional valence that even allowing the transgressive act to pass without comment provides a test of fealty, and out-group members may refuse to be party to the conversation.

This understanding of sociolinguistic signaling phenomena can serve to inform work on automated irony detection moving forward. For example, explicit modeling of the relationship(s) between speaker and audience (and distinguishing between in-group members and nonmembers) constitutes one promising direction to pursue.

6 Project Summary

As outlined in detail above, we have realized all project aims, thus making substantive progress on the very difficult task of recognizing verbal irony in online texts (posts). Our key argument throughout this project is that inferring irony requires a sociolinguistic context, and this had been missing from machine learning models for verbal irony detection. We have now shown that humans require contextualizing information to discern irony, and that standard token-based machine learning approaches misclassify many of the same comments for which annotators tend to request context [36]. Motivated by this finding, we then developed novel models that operationalize context in different ways [38, 35, 20]. These approaches consistently improved prediction performance, as desired.

To recapitulate: this grant has directly supported work that has culminated in four publications [36, 35, 38, 20], which have received in excess of 60 citations already. Together these works describe realizations all of the proposal objectives. Further, in the past year, PI Beaver has developed a new framework that we believe will lead to further progress on automated models for irony detection, again informed by sociolinguistics.

References

[1] Ramón Astudillo, Silvio Amir, Wang Ling, Mario Silva, and Isabel Trancoso. Learning word representations from scarce and noisy data with embedding subspaces. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1074–1084, Beijing, China, July 2015.

[2] David Bamman and Noah A Smith. Contextualized sarcasm detection on twitter. In *Ninth International AAAI Conference on Web and Social Media (ICWSM)*. Association for the Advancement of Artificial Intelligence, 2015.

[3] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[4] Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479, 1992.

[5] John D Campbell and Albert N Katz. Are there necessary conditions for inducing a sense of sarcastic irony? *Discourse Processes*, 49(6):459–480, 2012.

[6] P Carvalho, L Sarmento, MJ Silva, and E de Oliveira. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In *CIKM workshop on Topic-sentiment analysis for mass opinion*, pages 53–56. ACM, 2009.

[7] HH Clark and RJ Gerrig. On the pretense theory of irony. *Journal of Experimental Psychology*, 113:121–126, 1984.

[8] J Cohen. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*, 20:37–46, 1960.

[9] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537, 2011.

[10] D Davidov, O Tsur, and A Rappoport. Semi-supervised recognition of sarcastic sentences in twitter and amazon. pages 107–116, 2010.

[11] Chris Dyer. Notes on noise contrastive estimation and negative sampling. *arXiv preprint arXiv:1410.8251*, 2014.

[12] RW Gibbs and HL Colston. *Irony in language and thought: a cognitive science reader*. Lawrence Erlbaum, 2007.

[13] HP Grice. Logic and conversation. *1975*, pages 41–58, 1975.

[14] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[15] Vineet Kumar. Sarcasm detection: Beyond machine learning algorithms. In *Tiny Transactions on Computer Science*, 2015.

[16] Quoc V Le and Tomas Mikolov. Distributed representations of sentences and documents. *arXiv preprint arXiv:1405.4053*, 2014.

[17] Jiwei Li, Alan Ritter, and Dan Jurafsky. Learning multi-faceted representations of individuals from heterogeneous evidence using neural networks. *arXiv preprint arXiv:1510.05198*, 2015.

[18] S Lukin and M Walker. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. *NAACL*, pages 30–40, 2013.

[19] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[20] Silvio Moreira, Byron C. Wallace, Hao Lyu, Paula Carvalho, and Mário J. Gaspar da Silva. Modelling Context with User Embeddings for Sarcasm Detection in Social Media. In *Proceedings of the Conference on Computational Natural Language Learning (CoNLL)*, pages 167–177. SIGNLL, 2016.

[21] Frederic Morin and Yoshua Bengio. Hierarchical probabilistic neural network language model. In *Aistats*, volume 5, pages 246–252. Citeseer, 2005.

[22] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pages 807–814, 2010.

[23] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

[24] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

[25] E Riloff, A Qadir, P Surve, LD Silva, N Gilbert, and R Huang. Sarcasm as contrast between a positive sentiment and negative situation. In *EMNLP*, pages 704–714, 2013.

[26] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1, 1988.

[27] R Socher, A Perelygin, JY Wu, J Chuang, CD Manning, AY Ng, and C Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1631–1642. Citeseer, 2013.

[28] D Sperber and D Wilson. Irony and the use-mention distinction. *1981*, 1981.

[29] K Toutanova, D Klein, CD Manning, and Y Singer. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 173–180. Association for Computational Linguistics, 2003.

[30] O Tsur, D Davidov, and A Rappoport. ICWSM-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In *AAAI Conference on Weblogs and Social Media*, 2010.

[31] Y Tsuruoka, J Tsujii, and S Ananiadou. Stochastic gradient descent training for l1-regularized log-linear models with cumulative penalty. In *Proceedings of the Joint Conference of the Annual Meeting of the ACL and the International Joint Conference on Natural Language Processing of the AFNLP*, pages 477–485. Association for Computational Linguistics, 2009.

[32] AJ Viera and JM Garrett. Understanding interobserver agreement: the kappa statistic. *Family Medicine*, 37(5):360–363, 2005.

[33] BC Wallace. Computational irony: A survey and new perspectives. *Artificial Intelligence Review*, pages 1–17, 2013.

[34] BC Wallace and L Kertz. Can cognitive scientists help computers recognize irony? In *CogSci*, 2014.

[35] BC Wallace, DC Kook, and E Charniak. Sparse, Contextually Informed Models for Irony Detection: Exploiting User Communities, Entities and Sentiment. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1035–1044, Beijing, China, 2015. ACL.

[36] Byron C. Wallace, Do Kook Choe, Laura Kertz, and Eugene Charniak. Humans require context to infer ironic intent (so computers probably do, too). In *Proceedings of the Association for Computational Linguistics (ACL)*, pages 512–516. ACL, 2014.

[37] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

[38] Ye Zhang, Stephen Roller, and Byron C. Wallace. MGNC-CNN: A Simple Approach to Exploiting Multiple Word Embeddings for Sentence Classification. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL)*, page 1522–1527. ACL, 2016.

[39] Ye Zhang and Byron C. Wallace. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*, 2015.

[40] H Zou and T Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.